



HAL
open science

Les apports de l'apprentissage de la programmation par des situations en débranché au cycle 3 : quels transferts les élèves peuvent-ils faire d'une initiation à la programmation en débranché vers des activités sur un logiciel ?

Delphine Lamouroux

► To cite this version:

Delphine Lamouroux. Les apports de l'apprentissage de la programmation par des situations en débranché au cycle 3 : quels transferts les élèves peuvent-ils faire d'une initiation à la programmation en débranché vers des activités sur un logiciel?. Education. 2019. dumas-02418172

HAL Id: dumas-02418172

<https://dumas.ccsd.cnrs.fr/dumas-02418172v1>

Submitted on 18 Dec 2019

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

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

Master MEEF

« Métiers de l'Enseignement, de l'Éducation et de la Formation »

Mention second degré

Mémoire

Les apports de l'apprentissage de la programmation par des situations en débranché au cycle 3

**Quels transferts les élèves peuvent-ils faire d'une initiation à la
programmation en débranché vers des activités sur un logiciel ?**

Mémoire présenté en vue de l'obtention du grade de master

soutenu par

Delphine Lamouroux

le 22 mai 2019

en présence de la commission de soutenance composée de :

Christophe Declercq, directeur de mémoire

Sandrine Lemaire, membre de la commission

Remerciements

Je souhaite remercier toutes les personnes qui m'ont aidé lors de cette année de formation et lors de la rédaction de ce mémoire.

Dans un premier temps, je voudrais remercier mon directeur de mémoire, M. Declercq, pour sa disponibilité et ses conseils judicieux qui m'ont permis d'avancer dans ma réflexion et d'évoluer dans ma pratique de l'enseignement du numérique.

Je remercie l'équipe pédagogique de mon école d'affectation pour leur soutien, leurs précieux conseils et leurs encouragements. Je remercie également les élèves des deux classes avec qui j'ai mis en pratique l'enseignement du numérique et de la programmation et qui se sont montrés très coopératifs.

Je tiens à remercier mes collègues de formation avec qui j'ai échangé tout au long de l'année pour faire évoluer ma pratique et alimenter ma réflexion.

Je remercie également ma famille et mes amis pour leur soutien et leurs encouragements.

Sommaire

Remerciements	1
Sommaire	2
1. Introduction	4
2. Contexte théorique	6
2.1. L'informatique dans la recherche	6
2. 1.1. Le langage de la programmation.....	6
2. 1. 2. Les environnements de travail pour enseigner la programmation	6
2. 1. 3. L'informatique en débranché.....	7
2.2. L'analyse de l'activité de l'élève	8
2. 2. 1. Les capacités de l'élève face à la découverte d'un objet nouveau	8
2. 2. 2. Les postures de l'élève dans une activité de programmation.....	9
2.3. Cadre institutionnel.....	10
2. 3. 1. L'informatique dans l'enseignement à l'école primaire.....	10
2. 3. 2. La loi d'orientation et de programmation pour la refondation de l'Ecole de la République	11
2. 3. 3. Le socle commun de connaissances de compétence et de culture	11
2. 3. 4. Le lien entre le numérique et les mathématiques dans les programmes ...	12
3. Situation générale	13
3. 1. Cadre de la recherche	13
3. 1. 1. Questionnement.....	13
3. 1. 2. Hypothèses formulées	13
3. 2. Séquence détaillée	14
3. 2. 1. Séances d'apprentissage de la programmation en débranché	15
3. 2. 2. Choix du logiciel de travail	21
3. 2. 3. Séance de programmation branchée.....	23
3. 3. Protocole d'observation.....	24
3. 3. 1. Consignes de travail données par l'enseignant	25
3. 3. 2. Composition des binômes	25
3. 3. 3. Recueil de données.....	25
4. Analyse	27
4. 1. Comparaisons générales entre les deux groupes observés	27
4. 1. 1. Enrôlement dans l'activité.....	27

4. 1. 2. Avancée dans les deux classes observées	28
4. 1. 3. Répartition du travail au sein d'un binôme	29
4. 1. 4. Le conflit socio-cognitif.....	31
4. 1. 5. Gestion du temps	32
4. 2. Découverte du logiciel Pixel'Art.....	35
4. 2. 1. Lecture et compréhension des commandes	35
4. 2. 1. Essais du programme	37
4. 3. Usages du langage du codage.....	38
4. 3. 1. Les commandes « tourner à gauche » ; « tourner à droite ».....	38
4. 3. 2. La boucle de répétition.....	40
4. 4. Stratégies observées face à l'erreur	45
4. 4. 1. Réinitialisation du programme.....	45
4. 4. 2. Programmeur pas à pas	45
4. 4. 3. Changement de programmeur	46
4. 4. 4. Demande de solution à un tiers	46
5. Conclusion.....	47
6. Bibliographie	48
7. Annexes	50
Séquence détaillée	50

1. Introduction

Je suis en stage en classe à double niveau de CM1-CM2 dans une école élémentaire en milieu rural. Dans le cadre de l'enseignement de l'espace et de la géométrie, j'aborde le thème de la programmation avec les élèves. Il est inscrit dans les programmes en vigueur à la rentrée scolaire 2018 que les élèves doivent apprendre à programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation. La programmation peut se travailler en branché grâce à un logiciel de programmation ou en débranché par des activités amenant à coder des déplacements d'un personnage sans avoir recours à un outil numérique.

Pour travailler cette notion j'ai décidé de proposer aux élèves des activités en débranché et plus tard dans l'année, mener une séance de réinvestissement sur un outil numérique. Je souhaitais m'orienter vers des activités de programmation en débranché car il n'est pas toujours facile de disposer de suffisamment d'ordinateurs au cours d'une séance pour que chaque élève puisse être au travail. De plus, certaines études démontrent également que les enfants passent déjà beaucoup de temps devant un écran, bien qu'ici l'usage soit dans un but pédagogique, je souhaitais trouver une alternative qui permette aux élèves de travailler les mêmes compétences sans avoir la nécessité d'être sur un environnement numérique. Par la pratique d'activités en débranché, je souhaitais permettre aux élèves une première familiarisation avec le langage du codage en programmation mais également leur donner des méthodes de travail qu'ils pourront réinvestir lors d'une séance ultérieure sur un logiciel de programmation.

Je me suis renseignée sur la didactique de l'enseignement de la programmation en cycle 3. Je me suis interrogée sur les stratégies que pourrait adopter l'élève pour résoudre des situations de programmation proposées ainsi que les compétences qu'il mobilise lorsqu'il travaille sur une activité de programmation. Je me suis également questionnée sur le réinvestissement que les élèves de cycle 3 pourraient faire des apprentissages effectués sur des activités de programmation en débranché vers des activités sur un logiciel de programmation.

Pour répondre à cela, j'ai décidé d'observer deux groupes d'élèves de cycle 3 dans une activité de programmation sur un logiciel. L'un des deux groupes aurait au préalable été mis en

situation d'apprentissage par des activités de programmation en débranché alors que pour le second groupe, l'activité de programmation sur un logiciel serait une première séance dans ce domaine.

Dans une première partie je présenterai le cadre théorique de mon mémoire en présentant la notion de programmation et d'activités en débranché dans la recherche ainsi que les liens entre l'enseignement du numérique et les programmes de l'éducation nationale. Ensuite, j'exposerai le cadre de ma recherche, les hypothèses que j'ai formulées ainsi que les conditions que j'ai choisies d'appliquer dans les deux classes. Pour finir je présenterai et analyserai les données recueillies suite aux observations dans les deux classes de cycle 3 pour essayer de percevoir quels écarts il existe entre les deux groupes étudiés.

2. Contexte théorique

2.1. L'informatique dans la recherche

2.1.1. Le langage de la programmation

Le codage est un langage utilisé pour communiquer avec l'ordinateur. Il existe différents types de codage permettant différentes actions : la composition de structure de pages, l'animation d'éléments, le déplacement de personnages, etc. Dans l'apprentissage de la programmation à l'école, les élèves sont amenés à fréquenter un langage de programmation visuel simple présentant une syntaxe concrète de l'ordre qui sera exécuté pour faciliter leur compréhension et leur permettre une première approche du domaine de la programmation en informatique. Le langage étudié dans le cadre scolaire peut être composé de flèches indiquant des déplacements qui sera adapté à des élèves novices ou non lecteurs, mais il peut également être fait de courtes phrases exprimant une action précise tel que « avance », « recule », « tourne à droite »... Les ordres sont reliés par un système d'emboîtement, de flèches, ou de lignes qui permettent de visualiser le sens et le lien qui existe entre eux.

2.1.2. Les environnements de travail pour enseigner la programmation

Il existe beaucoup d'environnements différents permettant de travailler des activités de programmation. Des robots éducatifs programmables ont ainsi été créés comme les robots « Blue-Bot », « Bee-bot » ou encore « Thymio ». Ils permettent une première expérience ludique en programmation avec la possibilité de manipuler un objet numérique. La majorité de ces robots évoluent au sol à l'aide de touches disposées directement sur le dos de l'appareil. Les touches correspondent à des ordres basiques (avance, recule, tourne à gauche, tourne à droite) et sont symbolisés par des flèches permettant aux enfants de tout âge d'accéder au fonctionnement de cet outil de programmation. Grâce à des applications venant en complément de l'usage de ces robots, il est possible de faire une transition vers une forme différée de leur utilisation en passant d'une programmation en réel directement sur le robot vers une programmation virtuelle en créant un programme sur un outil séparé de l'engin.

Beaucoup de logiciels comme Scratch ou Scratch junior sont également adaptés à l'apprentissage de la programmation. Ils proposent des activités de déplacement de personnage ou de dessin de figures

par l'utilisation d'un langage visuel avec des ordres exposant clairement l'action qui sera effectuée. Il s'agit là de sélectionner des blocs de code et de les placer les uns à la suite des autres dans l'ordre désiré. En lançant l'exécution du programme, les actions s'effectueront selon le programme inscrit. Ces environnements sont davantage adaptés au cycle 3 puisqu'il peut être nécessaire de mobiliser des compétences d'anticipation du code.

Depuis 2011, le concours Castor est accessible aux élèves dès le cycle 3 jusqu'à la classe de terminale. L'objectif est de faire découvrir le numérique aux élèves. Le concours est composé de défis divisés en plusieurs niveaux de difficultés. Différentes compétences sont mobilisées y compris des compétences liées à la programmation comme le développement de la pensée algorithmique par l'anticipation d'un déplacement de personnages sur l'écran.

2. 1. 3. L'informatique en débranché

La pratique de l'informatique en débranché est née d'un programme d'enseignement appelé « computer science unplugged » créé par Tim Bell, Ian H Witten et Mike Fellows et par le concours de Robyn Adams et Jane McKenzie, il a été adapté à un usage en classe (consultable à l'adresse suivante : <https://csunplugged.org/en/>). L'objectif de ce programme est de proposer des situations d'enseignement apprentissage de l'informatique sans recourir à l'outil numérique comme l'ordinateur. Les auteurs considèrent en effet que l'usage de l'ordinateur n'est pas un élément indispensable pour acquérir des compétences en informatique et qu'il peut devenir une distraction nuisant aux apprentissages.

Lorsque l'on enseigne l'informatique par des activités en débranché, les situations proposées sont souvent simples et ludiques favorisant la manipulation par l'enfant. Cela permet à l'élève de s'engager plus rapidement dans la tâche. Les activités proposées en débranché permettent également de travailler la coopération entre pairs. Comme le précisent Georges-Louis Baron et Béatrice Drot-Delange dans un article de la Revue française de pédagogie (2016), « L'enjeu est de découvrir les concepts de l'informatique par des jeux fondés sur des situations pouvant mettre en oeuvre des objets concrets (...). C'est une pédagogie de découverte par l'activité. ». Les activités en débranchées peuvent constituer une première entrée dans l'initiation à l'informatique et peuvent également permettre de préparer les élèves à l'acquisition de compétences spécifiques au numérique.

Dans le domaine de l'apprentissage de la programmation, il existe de nombreuses activités qui permettent à l'élève de se familiariser avec le langage du codage et de commencer à développer des compétences d'anticipation. Un exemple est le « Robot Idiot » : une scène est symbolisée au sol dans la classe par un quadrillage. Un individu joue le rôle du robot, il doit se placer sur une case et attendre les instructions de déplacement que lui donnera une autre personne. Celui qui incarne le robot ne peut exécuter que des actions simples, il ne doit pas être dans l'interprétation des ordres qu'on lui donne. Le but de l'activité est de donner des consignes simples pour faire déplacer le robot d'un point de départ à un point d'arrivée. Dans cette activité, les participants s'approprient le langage du codage en interprétant des ordres précis. Ils apprennent également à changer de référentiel puisque pour guider le robot ils doivent se mettre à sa place pour donner le bon ordre, la droite du robot ne sera pas forcément la droite de la personne qui le guide.

D'autres supports ont été conçus pour permettre une pratique de la programmation en débranché. Par exemple, le jeu « Code Master : programming logic game » inventé par Mark Engelberg permet à l'élève d'apprendre les bases du codage en devant programmer le déplacement d'un personnage sur une scène. Ils se familiarisent également avec l'anticipation des actions d'un programme puisqu'ils doivent créer une séquence d'ordres avant d'exécuter le programme.

2.2. L'analyse de l'activité de l'élève

2. 2. 1. Les capacités de l'élève face à la découverte d'un objet nouveau

Pour analyser les comportements des élèves en situation d'apprentissage autour du numérique, Olivier Grugier (2018) a élaboré une grille d'analyse permettant de situer l'élève dans sa pratique.

On peut voir grâce à cette grille quelles sont les compétences développées par l'élève ainsi que les stratégies qu'il est capable de mettre en place pour se servir du matériel numérique mis à sa disposition. L'objectif de l'apprentissage de la programmation en cycle 3 est d'amener l'élève à entrer dans le registre de modification. L'élève devient capable d'utiliser les différentes fonctionnalités d'un objet de façon pertinente et d'optimiser son travail pour élaborer des stratégies adaptées aux situations présentées.

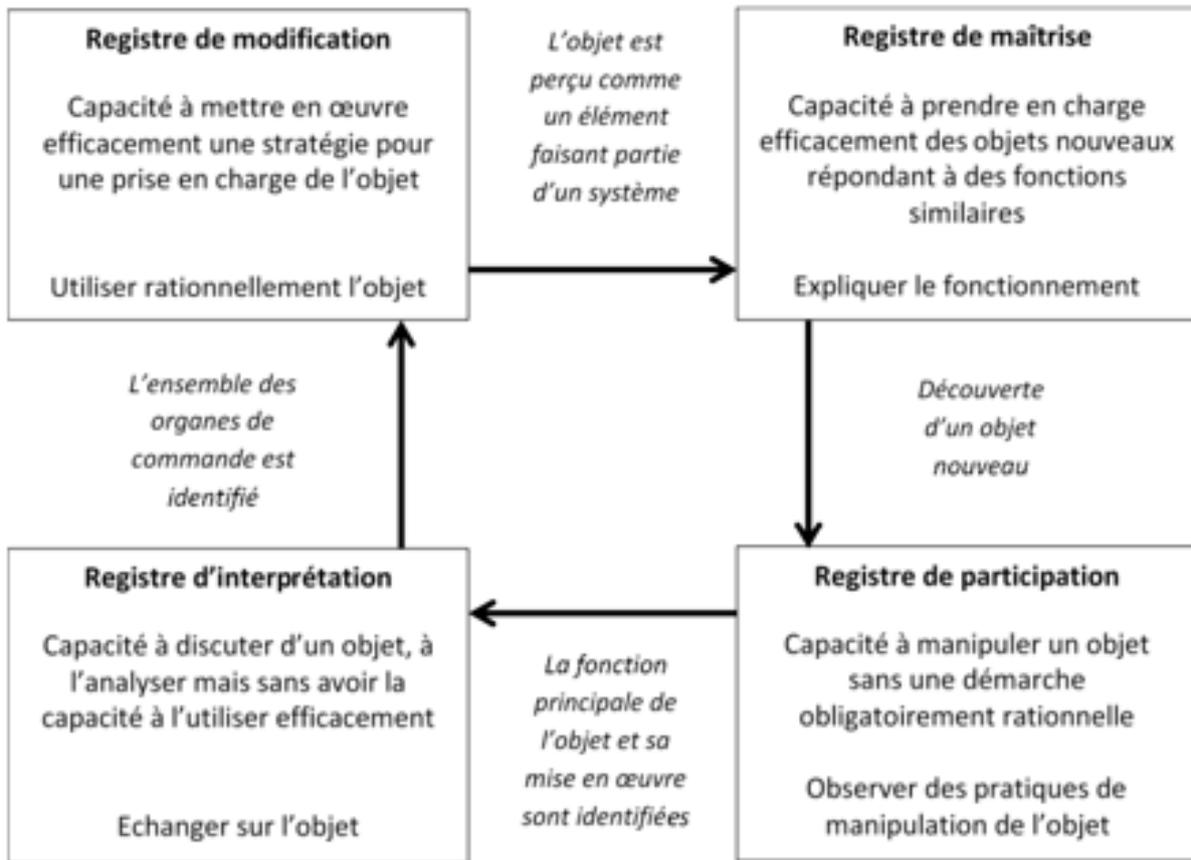


figure 1. Grille d'analyse de la technicité mise en jeu dans les moments d'enseignements-apprentissages proposés

2. 2. 2. Les postures de l'élève dans une activité de programmation

L'élève peut adopter différentes postures pour résoudre une situation dans laquelle il doit programmer le déplacement d'un personnage. Christophe Declercq et Françoise Tort (2018) ont analysé quatre postures pouvant être observées chez des élèves de cycle 3 en situation sur le logiciel de programmation Scratch.

La première posture est décrite ainsi : « l'utilisateur de la télécommande, effectue des manipulations directes mais ne programme pas ». L'élève donne une commande et celle-ci est immédiatement exécutée. Dans cette posture, l'élève cherche à déplacer immédiatement le personnage et il ne va pas anticiper plusieurs commandes de déplacement à l'avance.

La seconde posture est la suivante : « le programmeur pas à pas, peut réussir un programme simple mais ne développe pas de compétence d'anticipation ». L'élève va tester différentes commandes afin de voir quelle exécution est faite par le logiciel puis va les assembler pour construire le programme recherché. Il aura été capable de créer un programme simple sans avoir mis en jeu une compétence d'anticipation des commandes.

La troisième posture combine une utilisation de télécommande à la rédaction d'un programme : « le programmeur de télécommande est dans une posture de programmeur mais reporte la définition de la stratégie de résolution du problème, sur le futur utilisateur de la télécommande ». L'élève va écrire un programme qui associe un évènement précis à une touche du clavier. Ainsi il crée un programme qui permet de résoudre la situation en utilisant une télécommande.

La quatrième posture décrite est celle du programmeur. L'élève rédige un programme qui lorsqu'il sera exécuté permettra au personnage de se déplacer. L'élève aura mis en oeuvre des capacités d'anticipation du déplacement du personnage afin que lorsque le programme s'exécute, le personnage atteigne son but sans autre intervention de la part de l'élève. Cette posture, comme le précisent Christophe Declercq et Françoise Tort, est celle qui est attendue de l'élève dans le cadre de l'enseignement de la programmation.

2.3. Cadre institutionnel

2. 3. 1. L'informatique dans l'enseignement à l'école primaire

La vision de l'informatique dans l'éducation a beaucoup évolué depuis le rapport Simon de 1980 dans lequel il n'était pas envisagé d'introduire l'enseignement de l'informatique aux élèves avant la classe de 4ème comme le rappellent Georges-Louis Baron et Béatrice Drot-Delange (2016).

En 1985, le gouvernement met en place un programme : le plan d'informatique pour tous. Lors de la présentation à la presse, le 25 janvier 1985, il a été précisé l'importance de l'enseignement de l'informatique à l'école : « L'informatique va devenir de plus en plus une véritable seconde langue. L'objectif du Président de la République, le nôtre, est de faire de cette génération la mieux formée de notre histoire. Grâce à ce plan, la France va être dès cette année un des premiers pays du monde, probablement le premier, dans l'enseignement de l'informatique pour tous. »¹. Ce plan portait trois objectifs : l'initiation des élèves de toutes les régions à l'informatique et notamment à la programmation, permettre l'accès à l'outil informatique pour tous les citoyens et la formation des enseignants. Les écoles devaient être équipées de matériel informatique afin de répondre aux objectifs de ce plan. On constate l'importance qui était accordée par le gouvernement à former les élèves à l'usage du numérique lors de la publication de ce plan qui fera par la suite l'objet d'une circulaire n°85-0132 et n°85-208 du 29 mai 1985.

¹ Fabius, L. (1985).

Les circulaires qui paraissent les années suivantes précisent les objectifs de familiarisation à l'environnement informatique ainsi que l'apprentissage de l'usage du traitement de texte. L'usage de ces outils est perçu comme un biais pour accéder à des informations, il faut enseigner aux élèves l'utilisation de l'ordinateur pour construire leur savoir. Un enseignement développant des compétences méthodologiques sur l'utilisation des outils informatiques (ordinateurs et logiciels de traitement de texte, tableurs...) est alors préconisé.

Le BO n°42 du 23 novembre 2000 présente une nouvelle certification dont l'objectif est d'évaluer les compétences acquises et la maîtrise des outils informatiques par les élèves : il s'agit du brevet informatique et internet (B2i). Ce brevet comporte deux niveaux l'un ayant pour objet de « vérifier l'acquisition de compétences que les élèves peuvent maîtriser à l'issue de l'école primaire. » le second s'adressant aux élèves de collèges.

2. 3. 2. La loi d'orientation et de programmation pour la refondation de l'Ecole de la République

La loi n°2013-595 du 8 juillet 2013 d'orientation et de programmation pour la refondation de l'Ecole de la République porte parmi les mesures clefs celle de faire entrer l'École dans l'ère du numérique notamment par la mise en place d'une éducation au numérique. L'objectif est de mettre le numérique ainsi que les objets connectés au service des apprentissages et du développement des compétences numériques des élèves. Il est reconnu que les activités de programmation permettent également de développer des compétences dans la résolution de problèmes. Ce type d'activités permet également aux élèves de pratiquer et manipuler des outils ce qui rend l'apprentissage davantage motivant et concret.

2. 3. 3. Le socle commun de connaissances de compétence et de culture

L'enseignement du numérique s'inscrit dans plusieurs domaines du socle commun de connaissances de compétence et de culture.

Cet enseignement contribue au domaine 1 : *Les langages pour penser et communiquer* en travaillant l'item « Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques ». L'apprentissage des langages informatiques est un enjeu de ce domaine. Le langage du codage utilisé dans les activités de programmation fait l'objet d'un apprentissage. L'enseignement du numérique et plus précisément de la programmation s'inscrit également dans le

domaine 2 : *Les méthodes et outils pour apprendre* lorsque les élèves apprennent à utiliser des logiciels initiant à la programmation. Il est également fait mention de l'intérêt de l'utilisation d'outils numérique pour développer le travail collaboratif chez les élèves.

Dans le domaine 4 : *Les systèmes naturels et les systèmes techniques*, on précise également que l'usage d'outils numériques contribue à former les élèves à la démarche de résolution de problèmes dans le domaine des mathématiques.

L'utilisation des outils numériques est un enjeu du socle commun de connaissance, de compétence et de culture dans la formation des élèves. Il s'agit d'un outil au service des autres apprentissages et au service des compétences indispensables que l'élève doit avoir acquis à la fin de la scolarité obligatoire.

2. 3. 4. Le lien entre le numérique et les mathématiques dans les programmes

Dans les programmes du cycle 3 en vigueur pour l'année scolaire 2018-2019, le numérique ne constitue pas une discipline scolaire au même titre que l'enseignement du français ou des mathématiques. Il s'agit davantage d'un outil au service des disciplines enseignées à l'école primaire mais qui pourra être traduit par des compétences à travailler par les élèves.

Il est recommandé dans le domaine des mathématiques, de proposer aux élèves différents supports de travail incluant des outils numériques. Dans le domaine de l'espace et géométrie, l'usage de logiciels spécifiques est préconisé dans l'enseignement pour travailler la programmation de déplacements, la visualisation de cartes et de plans mais également la construction de figures. Sous la compétence « Reconnaître, nommer, décrire, reproduire, représenter, construire quelques solides et figures géométriques », il est recommandé dans les programmes de cycle de 3, de proposer aux élèves des situations de construction d'une figure plane à l'aide d'un logiciel de géométrie dynamique. Concernant l'apprentissage de la programmation, par la compétence « (Se) repérer et (se) déplacer dans l'espace en utilisant ou en élaborant des représentations », il est précisé que l'on attend des élèves qu'ils apprennent à « programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation ». L'outil numérique doit être mobilisé pour répondre à cette exigence des programmes.

3. Situation générale

3. 1. Cadre de la recherche

3. 1. 1. Questionnement

Suite aux lectures que j'ai pu faire, je me questionne sur les modalités de travail pour enseigner la programmation aux élèves de cycle 3. S'il est indiqué dans les programmes de cycle 3 en vigueur à la rentrée scolaire 2018 que les élèves doivent apprendre à programmer les déplacements d'un robot ou d'un personnage sur un écran, est-il judicieux d'exposer directement l'enfant au logiciel de programmation ? Je m'interroge sur l'intérêt que peut avoir la pratique de la programmation en débranché. Quels sont les impacts que peut avoir la pratique de la programmation et la familiarisation au langage du code par le biais d'activités débranchées ? Quels réinvestissements des élèves de cycle 3 peuvent-ils faire de l'apprentissage de la programmation en débranché vers une situation de travail branchée avec un logiciel ?

La question qui guidera mon recueil de données et mon analyse est la suivante : L'apprentissage de la programmation en cycle 3 par des activités débranchées permet-elle à l'élève d'acquérir avec plus d'aisance la compétence « Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation. » lorsqu'il sera confronté à des activités de programmation en branché ?

3. 1. 2. Hypothèses formulées

J'émetts l'hypothèse que le travail d'activité de programmation en débranché permet à l'élève d'acquérir des comportements qu'il pourra transposer aux activités branchés.

Je pense qu'un élève n'ayant pas fréquenté des activités lui permettant de se familiariser au langage du codage aura besoin d'un temps plus long de prise en main du logiciel. Il devra en effet découvrir un environnement numérique mais en plus comprendre un nouveau langage. Un élève ayant fait des activités de programmation de déplacements de personnages en débranché aura déjà été familiarisé au langage de la programmation, il sera capable d'entrer plus rapidement dans la tâche.

Je pose l'hypothèse qu'un élève ayant travaillé en débranché réinvestira également les éléments complexes du codage qu'il pourra avoir appris comme notamment l'usage de la boucle. Si c'est une notion qui est travaillée en débranché, je pense que l'élève l'utilisera lorsqu'il sera confronté à une situation de répétition alors que l'élève n'ayant pas fait de programmation en débranché conservera le simple usage des ordres de base sans utiliser de boucle de répétition.

Je suppose également que les élèves ayant travaillé sur des tâches de programmation d'un déplacement de personnage en débranché auront une meilleure capacité à anticiper les déplacements d'un personnage, ils auront moins recours à une vérification après chaque ordre ajouté à un programme et pourront l'écrire dans son intégralité (ou en grande partie) avant de le tester. Par la réciproque, je pense que les élèves n'ayant pas travaillé sur de la programmation procéderont davantage par tâtonnement pour résoudre les situations proposées, ils auront recours à une visualisation pas à pas et auront besoin de vérifier le programme qu'ils écrivent plus fréquemment que les autres élèves.

Grâce aux situations d'apprentissages en débranché, je pense également que les élèves auront plus de facilité à se répartir les rôles et à coopérer pour résoudre les problèmes qui leur seront posés. Ils se répartiront les différents rôles (programmeur qui dicte le programme à effectuer et l'exécutant qui applique les ordres du programme qui lui sont donnés) selon leurs compétences et les facilités qu'ils éprouvent dans l'un ou l'autre des rôles. Je pense également que si le programme qu'ils ont écrit comporte des erreurs, ils raisonneront sur ce qu'ils ont produit pour trouver où se trouve l'erreur alors que des élèves n'ayant pas été familiarisés à ces différents rôles auraient peut-être plus tendance à tout effacer pour recommencer sans raisonner sur le programme qui avait été fait.

[3. 2. Séquence détaillée](#)

La séquence se compose de 5 séances d'activités débranchées et d'une séance de réinvestissement sur une application Pixel Art.

3. 2. 1. Séances d'apprentissage de la programmation en débranché

- La première séance durait 50 minutes. Elle a eu lieu le 6 novembre. L'objectif de cette séance était que les élèves apprennent à se repérer sur un quadrillage. Les élèves étaient mis en activité autour du jeu de la bataille navale. Ils devaient placer des navires sur un quadrillage verticalement ou horizontalement. Puis, à tour de rôle il leur fallait indiquer à leur adversaire des cases sur lesquelles ils « tiraient » en donnant ses coordonnées. Le but étant de toucher tous les navires de l'adversaire avant lui. Cette séance permettait de travailler le repérage sur un quadrillage qui sera utile par la suite lorsqu'il faudra repérer un personnage ou d'autres éléments sur un quadrillage en programmation.

- La seconde séance s'est déroulée le 13 novembre d'une durée de 60 minutes. L'objectif de cette séance était d'accomplir et de décrire des déplacements dans l'espace familier en faisant du téléguidage dans un premier temps puis de la programmation. La situation proposée était celle du « robot idiot ». Un quadrillage est symbolisé par des plots disposés au sol dans la classe. Dans une case est disposée l'image d'un arc et dans une autre case l'image d'un monstre (voir figure 2).

Les élèves sont invités à s'installer tout autour de ce quadrillage. L'enseignant joue le rôle du robot.



figure 2. Quadrillage représenté par des plots au sol en fond de classe

Il se place sur une case et explique la situation aux élèves. « **Je dois aller affronter le monstre mais avant d'y aller, il faut que je récupère mon arc. Je dois me déplacer de case en case mais jamais en diagonale, guidez moi pour réussir ma mission.** » Les élèves prennent le rôle de programmeur, chacun leur tour, ils doivent donner un ordre pour guider l'enseignant qui ne fait

qu'obéir aux ordres. Ils sont dans la co-construction d'un programme. Les élèves comprennent vite que pour que le robot se déplace correctement sur le quadrillage, ils doivent prendre en compte sa position et son orientation afin d'énoncer les bonnes indications. Ils apprennent à se décentrer.

Lorsque la situation est résolue, un élève prend la place de l'enseignant et exécute les ordres donnés par ses camarades. L'exercice se répète ainsi plusieurs fois en changeant d'élève pour incarner le robot. Lors de cette situation, les élèves mettent en évidence qu'il est plus rapide d'énoncer l'ordre « *Avance de quatre cases* » plutôt que répéter quatre fois « *Avance* ». Ils cherchent à énoncer un programme plus rapide en donnant le moins d'ordre possible.

L'enseignant place ensuite un plot symbolisant le robot et montrant l'orientation du robot sur le quadrillage. Les élèves doivent retourner à leur place et reproduire le quadrillage sur leur cahier en plaçant le robot, le monstre et l'arc tels qu'ils sont placés dans le fond de la classe. Leur tâche est d'élaborer un programme permettant au robot de se saisir de l'arc puis de se rendre sur la case du monstre pour l'affronter. Les élèves doivent écrire les différents ordres qu'il faut donner au robot pour qu'il puisse accomplir sa mission (voir figure 3).

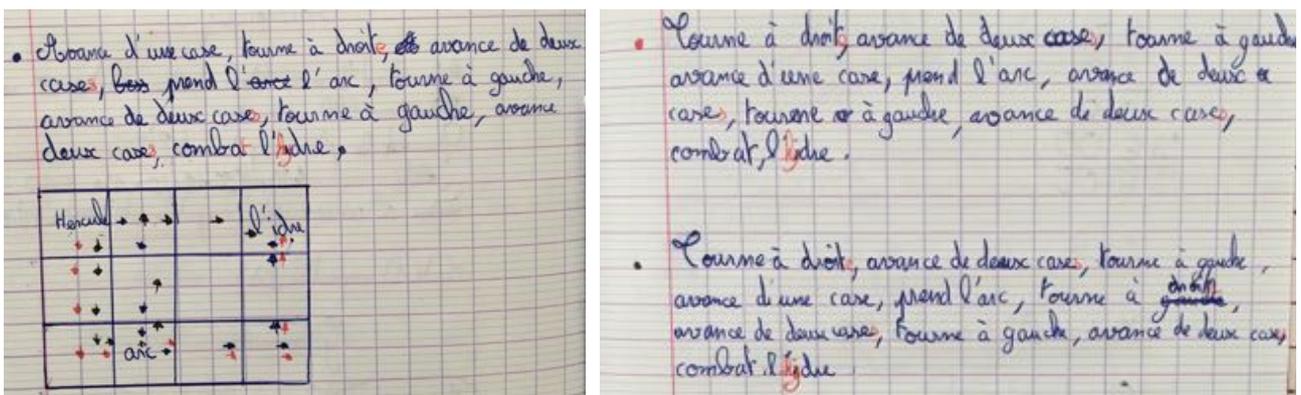


figure 3. Programmes rédigés par un élève pour déplacer le robot

Les élèves s'exercent ici à passer de déplacements dans l'espace réel à des déplacements dans l'espace relatif. Ils écrivaient leur programme à leur place mais avaient la possibilité de retourner en fond de classe en cas de doute ou pour tester le programme qu'ils avaient écrit. Lors de la mise en commun des programmes trouvés par les élèves, il est mis en évidence qu'il n'y a pas un seul chemin unique mais qu'il y a plusieurs possibilités qui permettent au robot d'accomplir sa tâche. Ils mettent en évidence qu'il y a des chemins plus longs que d'autres ou incluant davantage d'ordres à donner. Ils cherchent le chemin le plus direct pour résoudre la situation incluant le moins d'ordres à donner au robot.

- La troisième séance s'est déroulée le 19 novembre et durait 45 minutes, son objectif était de se repérer, décrire et exécuter des déplacements sur un plan. Les élèves devaient écrire le programme de déplacement d'un personnage sur un quadrillage. Ils n'avaient plus à leur disposition de représentation de la situation dans l'espace réel. Les élèves travaillent individuellement lors de cette séance sur des exercices disponibles dans leur manuel « *Le nouvel à portée de maths CM2* » (2016), n° 2 et n°3 à la page 115.

Dans l'exercice n°2, l'élève doit se repérer sur un quadrillage afin de placer des éléments en fonction de coordonnées données.

Dans l'exercice n°3, un déplacement est tracé sur le quadrillage, c'est à l'élève de coder le déplacement. La consigne de cet exercice amène les élèves à utiliser les coordonnées des cases pour coder le déplacement. Pour que les élèves réinvestissent les commandes qu'ils avaient vues lors des séances précédentes, j'ai fait le choix de modifier la consigne de cet exercice. Les élèves ne devaient plus indiquer les coordonnées de toutes les cases traversées comme demandé dans le manuel mais de coder un déplacement en utilisant le langage de programmation vu dans la séance précédente en utilisant les mots « avance » « tourne à gauche » « tourne à droite ». Ainsi, ils mobilisaient le langage du codage en continuant de travailler sur la décentration puisqu'ils devaient être capable de se mettre à la place d'un personnage orienté.

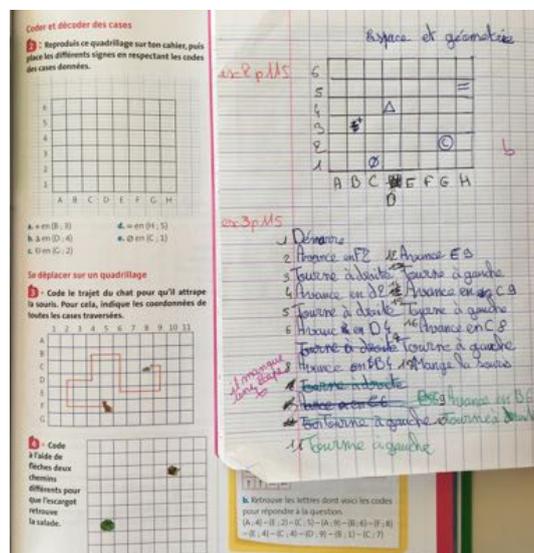
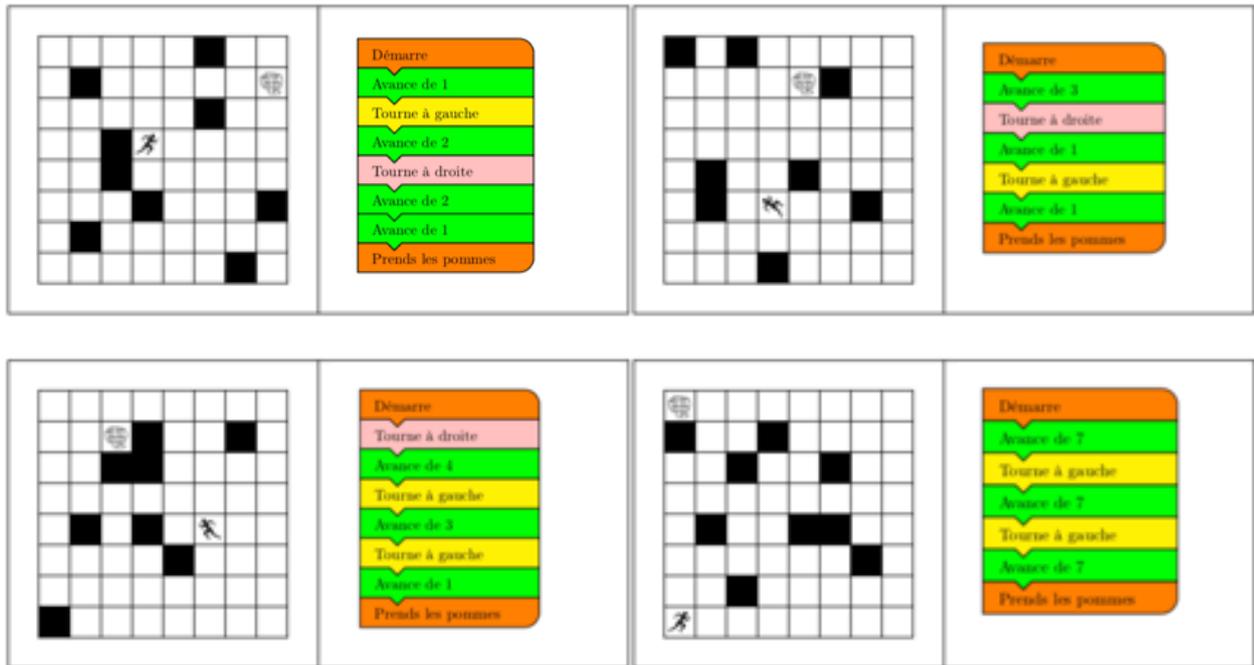


figure 4. Exercices réalisés par les élèves lors de la séance 3

- La quatrième séance a eu lieu le 20 novembre sur un créneau de 30 minutes. L'objectif était de programmer et anticiper les déplacements d'un personnage sur un quadrillage. Les élèves étaient en groupe de 4. Chaque groupe disposait de quatre quadrillages « scènes » et de quatre feuilles de

codes (Daval, 2017)² (voir figure 5). Sur chaque scène, on retrouvait les mêmes éléments : un personnage et un pommier mais disposés à différents endroits sur le quadrillage. La tâche des élèves était de faire correspondre une feuille de code à un quadrillage de façon à ce qu'en appliquant les ordres donnés, le personnage attrape une pomme dans le pommier.

Dominogrammes jaunes 1 à 4



N. Daval

ESPE Réunion, 2017

figure 5 Scènes et feuilles de code à appairer

Lors de cette séance, les élèves ont eu des difficultés à interpréter la position du personnage et les déplacements qu'il pouvait effectuer sur le quadrillage. Ils se servaient parfois de leur doigt qu'ils posaient à la place du personnage mais en cours de programme, ils pivotaient leur doigt et ne savaient plus dans quel sens était orienté le personnage. Certains ont également utilisé leur matériel scolaire comme leur tube de colle mais ils n'avaient pas pris d'indice sur le tube afin de pouvoir suivre l'orientation du personnage en cours d'exécution du programme.

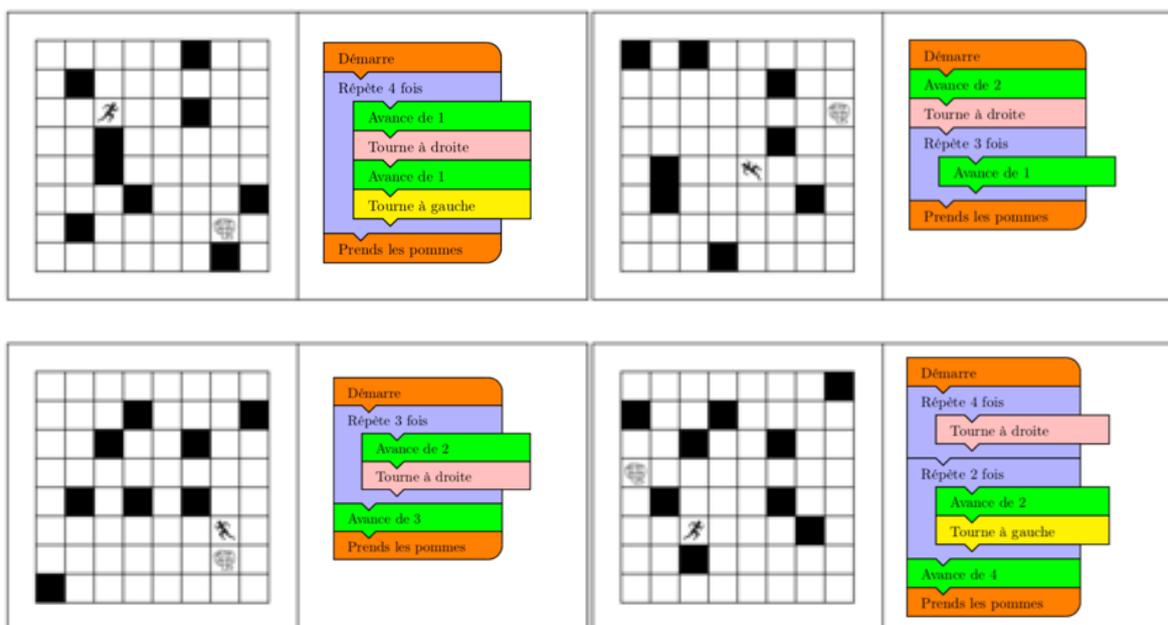
J'ai observé sur cette séance plusieurs stratégies mises en oeuvre par les élèves. Dans un groupe, chaque élève prenait une scène ainsi qu'une feuille de code et testait individuellement le programme pour voir si les deux feuilles correspondaient. Si ce n'était pas le cas, ils replaçaient leur feuille au milieu de la table et se saisissaient de deux autres feuilles pour tester à nouveau un autre programme sur une autre scène. Ils ne sont pas parvenus à la fin de la séance à faire correspondre tous les programmes à une scène à cause de cette organisation. Ils ne se concertaient pas entre eux sur ce

² Daval, N. (2017). *Dominogrammes jaunes*. disponible sur : <https://www.edumooov.com/fiche-de-preparation-sequence/83703/espace-et-geometrie/cm1-cm2/codage-et-programmation>

qu'ils avaient testés et sans s'en rendre compte, il leur arrivait de tester plusieurs fois la même chose. Dans un autre groupe observé, les élèves ont organisé leur travail ainsi : une élève se saisissait d'une scène et plaçait son doigt à la place du personnage dessiné sur la scène, une seconde élève avait une feuille de code, elle dictait les ordres à la première élève qui déplaçait son doigt sur la scène et validait ou non la réussite du programme. Une troisième élève s'était emparée de toutes les feuilles de codes, elle mettait de côté les feuilles de code déjà testées et qui ne fonctionnait pas avec la scène utilisée par la première élève et était chargée de donner au fur et à mesure les feuilles de code à la deuxième élève. Le quatrième élève conservait les paires déjà formées ainsi que les autres scènes qui n'étaient pas encore appareillées. Cette stratégie s'est avérée efficace pour les élèves qui ont formés toutes les paires. A la fin de la séance, les élèves de ce second groupe ont été invités à présenter leur stratégie et leur organisation au reste de la classe afin que les autres élèves puissent s'en emparer lors de la séance suivante.

- La cinquième séance a duré 55 minutes et s'est déroulée le 27 novembre. L'activité proposée était similaire à celle de la quatrième séance à la différence que les codes incluaient désormais des boucles de répétition. Une boucle permet de répéter une même action ou une même séquence d'actions un certain nombre de fois, les ordres à répéter sont glissés dans cette boucle. Le support est similaire à celui de la séance précédente (Daval, 2017)³ (voir figure 6).

Dominogrammes rouges 1 à 4



N. Daval

ESPE Réunion, 2017

figure 6. Scènes et feuilles de code à appairer

³ Daval, N. (2017). *Dominogrammes rouges*. disponible sur : <https://www.edumooov.com/fiche-de-preparation-sequence/83703/espace-et-geometrie/cm1-cm2/codage-et-programmation>

L'objectif était comme pour la séance précédente, d'anticiper et programmer les déplacements d'un personnage sur un quadrillage en ajoutant la notion de boucle et de répétition. Les élèves ont du interpréter les boucles en se questionnant sur leur sens et comment il fallait l'appliquer.

Suite aux difficultés rencontrées par les élèves lors de la séance précédente, j'ai adapté les supports. Chaque groupe avait à disposition un quadrillage d'aide (Daval, 2017)⁴ plus grand ainsi qu'une étiquette à placer représentant le pommier et la figurine d'un petit personnage orienté (voir figure 7).

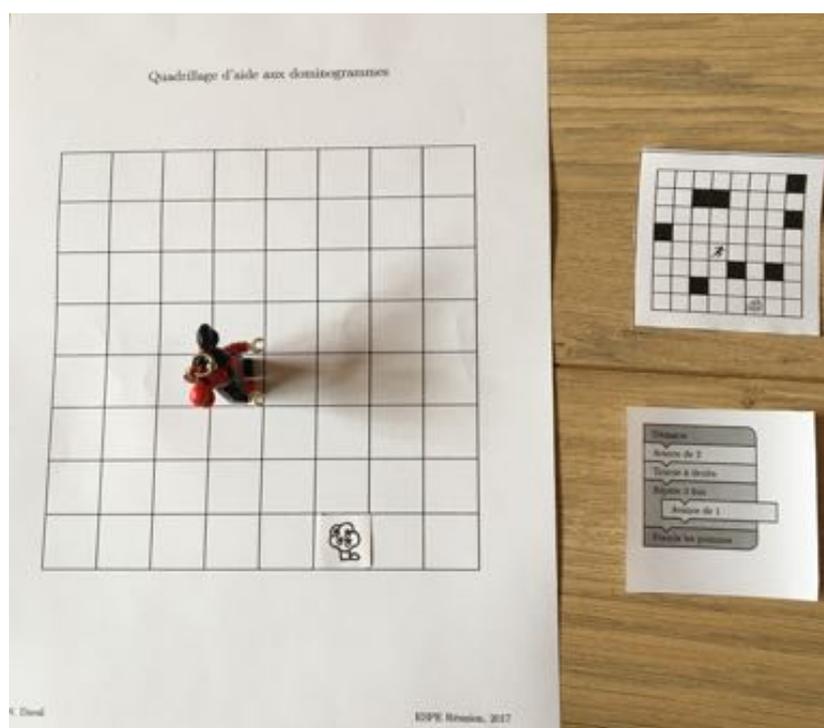


figure 7. Supports utilisés pour tester les situations

Ils devaient placer le pommier et la figurine sur le quadrillage en fonction de la scène qu'ils voulaient jouer. Ils pouvaient ainsi se repérer sur un quadrillage plus grand et plus lisible, ils pouvaient également directement déplacer le personnage orienté pour tester le programme. N'ayant les supports agrandis qu'en un seul exemplaire par groupe, les élèves devaient coopérer et résoudre ensemble les situations. Ils n'avaient plus la possibilité de tester chacun individuellement sans se concerter et prendre en compte le travail des autres membres du groupe. Le groupe qui avait eu des difficultés lors de la séance précédente s'est emparé de la stratégie qui leur avait été présentée. Ils ont été beaucoup plus organisé dans leur travail et sont parvenus à appairer les scènes avec les bons programmes.

⁴ Daval, N. (2017). *Quadrillage d'aide aux dominogrammes*. disponible sur : <https://www.edumoov.com/fiche-de-preparation-sequence/83703/espace-et-geometrie/cm1-cm2/codage-et-programmation>

3. 2. 2. Choix du logiciel de travail

Dans un premier temps, je souhaitais utiliser le logiciel recommandé par le manuel de la classe⁵ : Géotortue (voir figure 8). En observant l'environnement de travail du logiciel, j'ai constaté que les commandes proposées pour déplacer le personnage présentaient quelques difficultés. Ce sont des abréviations de l'ordre qui peut être donné : « av » pour avance, « re » pour recule, « tg » pour tourne à gauche et « td » pour tourne à droite. Après chaque abréviation un nombre est donné, ce nombre correspond à une mesure dont l'unité n'est pas donnée. Pour la commande « av » et « re » on suppose qu'il s'agit d'une longueur mais on ne peut affirmer s'il s'agit d'une mesure en millimètres ou en nombre de pixels. Pour les commandes « tg » et « td » un nombre est également affiché à côté sans unité, on peut déduire qu'il s'agit d'une mesure d'angle qui servira à faire pivoter le personnage.

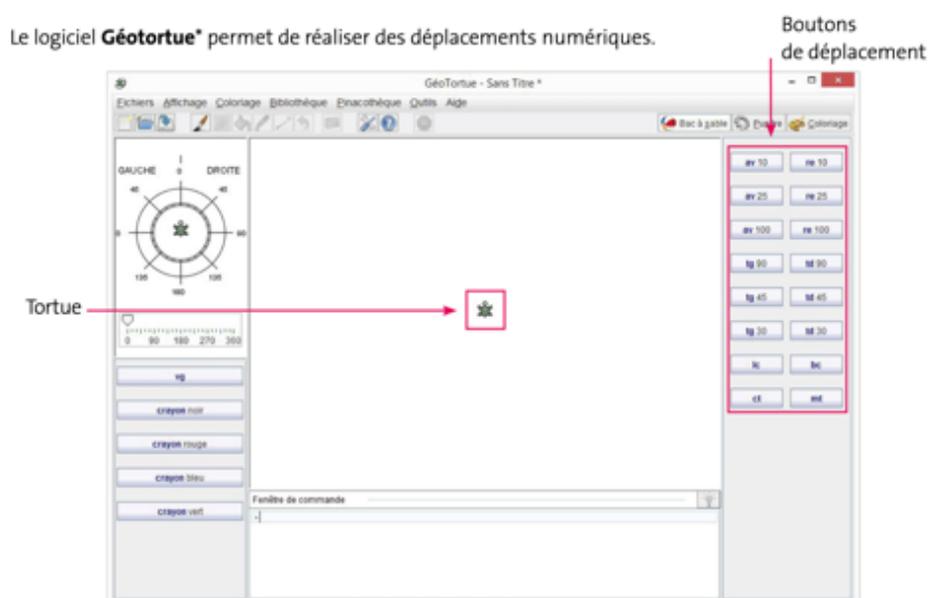


figure 8. Présentation de l'environnement de travail du logiciel Géotortue

Cependant pour l'élève, ce changement d'unité n'est pas visible car aucune unité de mesure n'est affichée à l'écran. Cela peut constituer un réel obstacle à la compréhension des commandes qu'il a à sa disposition. Les apprentissages liés aux angles sont initiés au cycle 3 mais le recours à la comparaison ne se fait pas par la mesure mais par superposition ou avec un calque. Utiliser des mesures d'angles pour orienter un personnage sur un logiciel peut constituer un obstacle pour les élèves. De plus, la feuille de dessin du logiciel Géotortue est complètement blanche. Dans les situations d'apprentissages de la programmation en débranché, les élèves ont travaillé avec des scènes quadrillées. Devoir se repérer sur une feuille blanche constituerait une autre difficulté pour

⁵ Lucas, J., Lucas, J.-C., Meunier, L., Trossevin, M.-P., Meunier, R. (2016) « Le nouvel à portée de Maths CM1 », p.118

les élèves. Les situations que proposent ce logiciel ne me semblent pas adaptées pour réinvestir les apprentissages qu'auront faits les élèves lors des séances d'enseignement-apprentissage de la programmation en débranché.

Je me suis interrogée si l'usage de Scratch Junior pouvait être pertinent à l'issue de la séquence. Cependant, comme les élèves avaient été familiarisés avec l'usage de lignes de codes écrites, il aurait été dommage de revenir à l'usage de simples flèches pour coder les déplacements. Cela ne leur permettait pas de réinvestir les ordres qu'ils avaient appris en langage de code.

J'ai finalement fait le choix de proposer une situation d'apprentissage sur le logiciel de programmation Pixel Art développé par Christophe Declercq et Michael Zeyringer (2018). Cet environnement a été élaboré avec pour objectif de conduire l'élève à adopter la posture de programmeur. Les différentes activités proposées consistent à reproduire un dessin sur un quadrillage en déplaçant une flèche (voir figure 9).



figure 9. Environnement de travail du logiciel Pixel'Art

Les lignes de code proposées reprennent les codes déjà appris par les élèves au cours de la séquence, ainsi je peux m'assurer que la compréhension du langage ne sera pas un obstacle pour les élèves ayant travaillé sur des activités en débranché. L'environnement de travail de ce logiciel est suffisamment épuré pour ne pas surcharger l'écran de commandes et d'informations. L'élève peut facilement se repérer entre les différents codes qu'il peut utiliser et les commandes de dessins.

De plus, les élèves ont travaillé sur des situations de codage impliquant des quadrillages similaires à la grille proposée sur l'application. Les élèves devraient avoir plus de facilité à se repérer sur ce quadrillage pour déplacer la flèche et créer le motif désiré.

Le logiciel permet également un retour instrumental à l'élève par les options « voir la trace » qui permet de constater de l'orientation et du déplacement de la flèche dans l'exécution du programme et « pas à pas » qui permet de voir la succession des instructions qui permet de mettre en lien le programme écrit et l'exécution qui en est faite.

Le choix de travailler sur un logiciel de Pixel Art permet de mobiliser l'intérêt des élèves. En effet, j'ai constaté qu'ils faisaient beaucoup de dessin à la main en Pixel Art sur des feuilles quadrillées. J'émet l'hypothèse que cela sera source de motivation pour eux d'avoir l'occasion de pratiquer cette activité sur un nouveau support et que cela faciliterait l'enrôlement des élèves dans l'activité de programmation proposée.

De plus, contrairement à Géo Tortue, les activités du logiciel Pixel'Art ne se limite pas à la programmation d'un déplacement d'un personnage sur un écran mais présentent un objectif concret pour les élèves : reproduire un dessin, un motif ce qui pourra être source de motivation chez certains élèves et favorisera l'enrôlement autour de la tâche proposée.

3. 2. 3. Séance de programmation branchée

- Une séance de réinvestissement sur l'outil numérique est prévue en période 3, le 28 janvier soit deux mois après la dernière séance en débranché. L'objectif est de transférer les apprentissages des élèves des activités débranchées vers une activité en branché. Entre la dernière séance de la période 2 et cette séance de réinvestissement, aucun rappel n'est fait auprès des élèves. J'ai fait ce choix pour pouvoir observer quels apprentissages les élèves ont conservés des activités débranchées et savoir ceux qu'ils réinvestissent dans une activité sur le numérique.

L'activité proposée aux élèves est sur ordinateur. Les élèves travaillent sur un logiciel de Pixel Art. Un quadrillage avec un motif de couleur est présenté aux élèves. Ils doivent reproduire ce motif dans un quadrillage vierge en déplaçant une flèche. Pour déplacer la flèche et colorer les cases, les élèves ont des lignes de codes à sélectionner et positionner dans un ordre permettant de reproduire le motif lors de l'exécution du programme. Une succession de 9 activités est proposée aux élèves avec une difficulté croissante (voir figure 10).

Les trois premières activités permettent une prise en main du logiciel et des commandes de base pour déplacer la flèche sur le quadrillage: « avance » « tourne DROITE » « tourne GAUCHE »

ainsi que les commandes de couleurs « rouge » ; « jaune » ; « vert » et « bleu ». Les activités 4, 5, 6, 7, 8 et 9 proposent une difficulté croissante. Le motif à reproduire implique davantage de commandes. C'est dans ces activités qu'il peut être judicieux de recourir à l'usage de la boucle puisque le motif présente un caractère répétitif dans les commandes à utiliser : répétition d'un enchaînement de couleurs sur une ligne, carré à reproduire dont les bords présentent une même combinaison de couleurs.

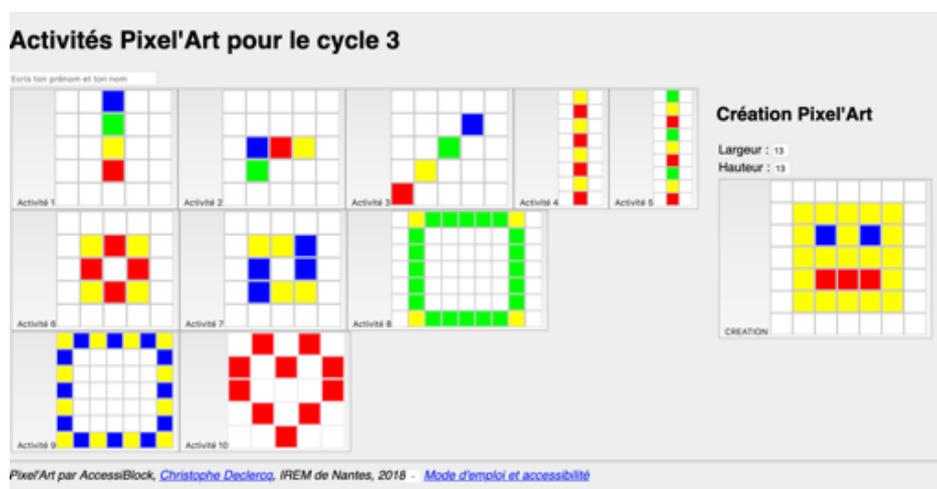


figure 10. Activités proposées sur le logiciel Pixel'Art

Lorsque les élèves ont terminé les activités proposées, ils ont la possibilité de faire une activité bonus : l'activité n°10 dans laquelle ils doivent reproduire le motif d'un coeur, ils peuvent également créer leur propre motif grâce à une activité créative sans motif imposé.

[3. 3. Protocole d'observation](#)

J'ai élaboré une séquence ayant pour objectif de développer l'anticipation d'un programme chez les élèves en les familiarisant avec le langage du codage par des activités de programmation en débranché. Cette séquence a été menée sur la période 2 de l'année scolaire 2018-2019 dans la classe de CM1-CM2 dans laquelle je suis affectée. La dernière séance de cette séquence consistera à programmer les déplacements d'un personnage sur un logiciel de programmation. J'ai également mené cette séance avec une autre classe de CM2 n'ayant pas travaillé la programmation au cours de cette année scolaire. J'ai mis en place un protocole d'observation de l'activité des élèves sur cette séance de réinvestissement afin de pouvoir observer et comparer leurs méthodes de travail. Je souhaite par cela observer les écarts et mesurer les apports de l'apprentissage de la programmation par des activités débranchées.

3. 3. 1. Consignes de travail données par l'enseignant

La situation est présentée de façon similaire aux deux classes. Les consignes données sont les mêmes. Les élèves ont la possibilité de poser des questions après la passation de consigne.

Consigne donnée : « **L'objectif de l'activité est de reproduire un motif en pixel art grâce à un logiciel. Un modèle vous sera présenté, vous devrez le reproduire sur un quadrillage vierge. Il y a plusieurs modèles à réaliser. Les activités doivent être faites dans l'ordre. Lorsque vous avez fini un motif, il faut appeler l'enseignant pour qu'il valide votre travail. Vous travaillerez en binôme, à vous d'organiser votre travail.** »

Pour favoriser les interactions, les élèves travaillent par binôme sur un seul ordinateur portable disposé entre les deux élèves. Ils sont libres de s'organiser comme ils le souhaitent pour réaliser le travail.

De l'aide a été apportée ponctuellement aux élèves rencontrant des difficultés. Je leur proposait de cocher les cases « pas à pas » et « voir la trace » avant de dessiner leur programme. Lorsqu'ils étaient bloqués sur un dessin qui ne s'affichait pas comme ils le pensaient, je leur demandais de verbaliser le programme qu'ils avaient rédigé dans le but de les faire réfléchir à ce qu'ils écrivaient dans le programme.

3. 3. 2. Composition des binômes

11 binômes ont été observés dans la classe de CM1-CM2 et 14 binômes dans la classe de CM2. Dans les deux classes, les élèves travaillent en binôme avec leur voisin de classe sans modification du plan de classe établi afin que les deux groupes observés soient dans des conditions similaires.

Dans la classe à double niveaux, les élèves sont répartis dans des binômes séparés entre les CM1 et les CM2. La seconde classe est composée uniquement d'élèves de CM2 dont le niveau d'aisance en activité de programmation est inconnue. Le choix des binômes à filmer est aléatoire.

3. 3. 3. Recueil de données

Lors des deux séances, j'ai adopté une position d'observateur afin de voir la diversité des situations et des procédures utilisées par les élèves. J'ai pris en note des bribes de conversations en

indiquant les interlocuteurs et l'heure à laquelle cela a été dit ainsi que l'activité en cours. J'ai essayé de décrire par écrit quelques interactions (impliquant ou non un échange verbal).

Le logiciel Pixel'Art permet de garder trace des manipulations faites par les élèves. A la fin de la séance, il m'a été possible de recueillir les traces de leur activité sous forme de fichier texte. J'ai pu mettre en lien les activités réalisées avec les commandes utilisées et l'heure précise de chaque action sur le logiciel. Cela permet de donner des indications sur leurs procédures pour résoudre l'activité proposée : s'ils ont procédé par tâtonnement en testant leur programme à chaque ligne de code ajoutée ou s'ils ont entré toutes les lignes de codes avant de tester le programme. Le délai entre l'utilisation des commandes permet d'avoir une idée du temps qu'il a fallu au binôme pour résoudre une situation proposée par le logiciel.

Dans chaque classe, un binôme a été filmé lors de la séance de programmation sur ordinateur. La caméra était placée derrière les élèves afin de capter leurs échanges mais aussi ce qu'ils faisaient au même moment sur l'ordinateur. Ils étaient avertis qu'ils étaient filmés pendant l'activité mais en plaçant la caméra derrière eux, on espère également qu'ils ont réussi à faire abstraction de cela et qu'ils sont restés spontanés dans leurs interactions sans que cela ne perturbe leur travail.

4. Analyse

4. 1. Comparaisons générales entre les deux groupes observés

4. 1. 1. Enrôlement dans l'activité

L'ensemble des élèves de CM1-CM2 est rapidement entré dans l'activité. Je pense que le fait d'avoir fait des activités de programmation en débranché facilitait l'enrôlement des élèves dans la tâche et leur donnait davantage envie de s'investir dans le travail à réaliser car ils avaient déjà des stratégies de travail qu'ils pouvaient réinvestir. Ils prenaient le temps d'analyser la figure à reproduire pour anticiper le programme qu'ils allaient devoir écrire. Les CM2 sont aussi rapidement entrés dans la tâche mais ils n'avaient pas le même investissement dans le travail en binôme que la classe ayant pratiqué la programmation en débranché. En effet, j'ai observé que sur l'ensemble des élèves, plus de la moitié ne restaient pas avec leur binôme lorsque c'était à l'autre de manipuler. Ils se déplaçaient beaucoup dans la classe pour aller observer le travail de leur camarade contrairement aux CM1-CM2 qui restaient à leur poste de travail. Ils allaient échanger avec leurs camarades en comparant leurs avancées dans les activités proposées. Un seul binôme ne s'est jamais déplacé pour aller voir le travail des autres pendant l'activité, ce binôme est également le seul à ne pas avoir dépassé l'activité 2.

En début de séance, un élève de la classe de CM2 m'a fait la remarque que « ce n'est pas du vrai travail, c'est du dessin sur ordinateur ! ». Il n'a pas perçu les compétences mathématiques de programmation qui étaient travaillées dans cette séance. Certains élèves ont pris la séance pour une séance de « jeu » c'est aussi pour cela qu'ils s'autorisaient à se déplacer librement dans la classe sans rester à leur poste avec leur binôme. Je n'ai pas entendu cette remarque dans l'autre classe. Ayant travaillé en débranché, les élèves de CM1-CM2 ont trouvé la situation ludique mais percevaient davantage les objectifs de travail. J'ai pu constater que plusieurs binômes de CM1-CM2 prenaient le temps de comprendre l'usage de la boucle, j'en déduis qu'ils ne restaient pas sur l'objectif de reproduire un motif mais souhaitaient écrire un programme performant en optimisant les ordres utilisés.

Les CM2 étaient également davantage dans le défi avec la volonté d'être plus rapide que les autres binômes de la classe ce qui n'a pas été le cas des élèves de CM1-CM2. Ils souhaitaient résoudre au plus vite les situations proposées et ne cherchaient pas à optimiser le code inscrit. Le binôme de la classe de CM2 qui a été le plus loin dans les activités proposées n'a jamais fait usage

de la boucle. Ils ne testaient leur programme qu'après avoir inscrit tous les ordres. En les observant, j'ai constaté qu'ils souhaitent aller au plus vite ce qui ne les amenait pas à réfléchir sur la notion de répétition. Ils se servaient uniquement des ordres dont ils avaient compris le fonctionnement et restaient ainsi dans leur zone de confort. Malgré leur rapidité dans la résolution des tâches, ils sont restés dans un registre d'interprétation sans être capable d'utiliser de façon efficace toutes les commandes mises à leur disposition.

4. 1. 2. Avancée dans les deux classes observées

Un seul binôme de la classe de CM1-CM2 est allé jusqu'à la réalisation de l'activité 10, il était composé de deux élèves de niveau CM2. La plupart des binômes étaient arrivés à l'activité 8 ou 9. Le binôme le moins avancé était composé de deux CM1 et se sont arrêtés après la réalisation de l'activité 6. Un binôme de CM2 est allé plus loin dans les activités que l'ensemble des élèves de CM1-CM2. Ils ont réalisé avec succès l'ensemble des activités proposées et ont également commencé un dessin libre. Deux binômes avaient atteint l'activité 10. Le binôme le moins avancé n'avait pas réussi à résoudre l'activité 2.

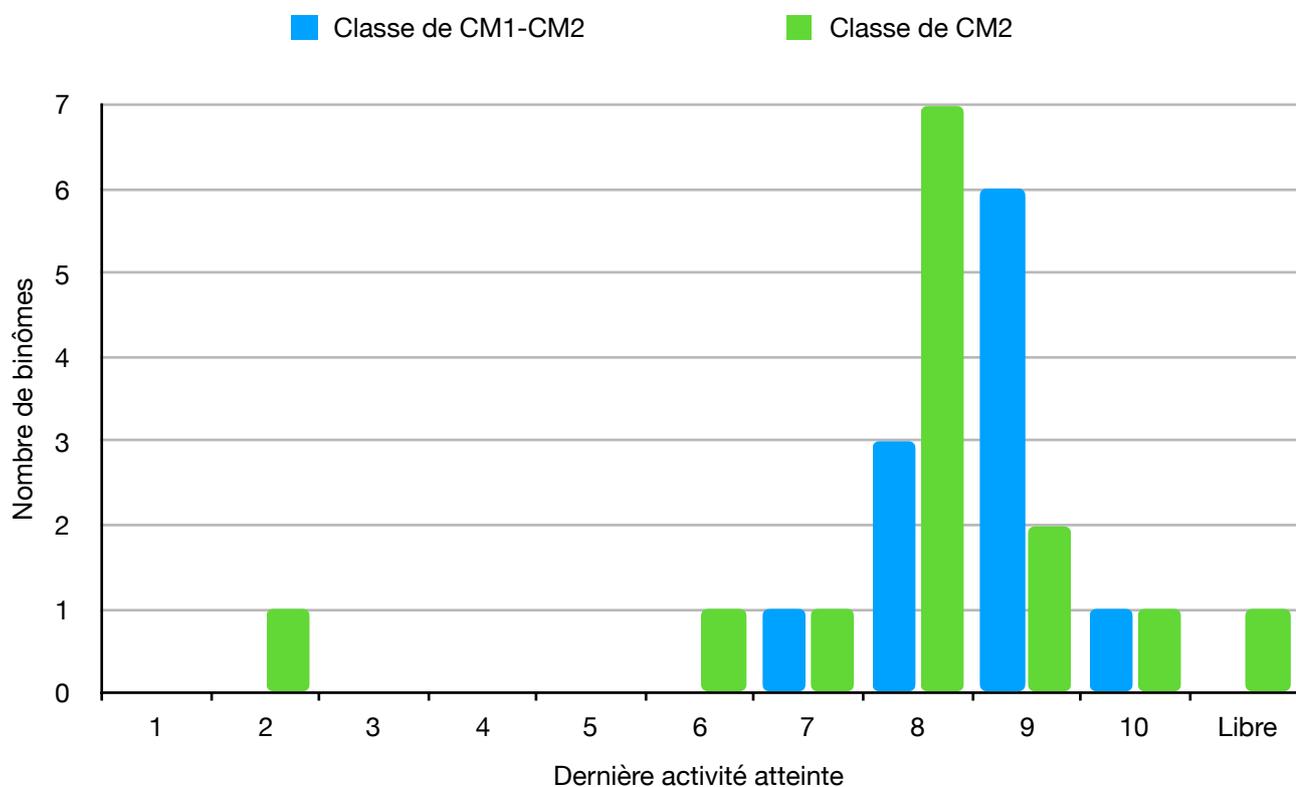


figure 11. Dernière activité atteinte par les binômes en fonction de la classe des élèves

La pratique d'activités débranché a donc permis d'avoir un niveau plus homogène lors de la mise en situation sur un logiciel de programmation. Les élèves ont dépassé certaines difficultés qu'ils ont rencontrées dans la programmation du déplacement d'un personnage ce qui leur a permis d'aller plus loin dans les activités en branché. L'apprentissage par des séances de pratiques débranchées a permis à l'activité sur le logiciel de programmation de se situer dans la Zone Proximale de Développement (Vygotski) de tous les élèves de cette classe. Aucun binôme ne s'est retrouvé bloqué. Les élèves rencontraient quelques obstacles dans l'usage de certaines commandes mais étant capables de les surmonter en raisonnant et en mobilisant leurs connaissances antérieures. A l'inverse, les élèves de CM2 rencontraient pour la première fois les difficultés pouvant être rencontrées dans ce type d'activité (compréhension du langage du codage, changement de référentiel pour orienter et déplacer le personnage à l'écran, anticipation des actions réalisées), certains n'ont pas réussi à surmonter ces difficultés et sont restés dans le registre de participation comme le définit O. Grugier. Ils manipulaient en ajoutant des ordres sur la feuille de programme mais sans adopter une démarche efficace pour résoudre le problème, je leur ai conseillé de tester successivement les différents ordres pour constater de leur effet sur le déplacement de la flèche, d'utiliser la commande « pas à pas » et « voir la trace » mais ils ne sont pas parvenus à réussir l'activité 2. Cela montre que les activités proposées étaient hors de leur zone proximale de développement puisque malgré les aides et conseils que je leur apportais, ils ne parvenaient pas à réussir l'activité proposée.

4. 1. 3. Répartition du travail au sein d'un binôme

Dans les deux classes, les élèves travaillaient en binôme entre voisins de classe. Ils étaient libre d'organiser leur travail comme ils le souhaitaient.

Dans le groupe classe de CM1-CM2 ayant travaillé auparavant sur des activités de programmation en débranché, j'ai constaté plusieurs méthodes de travail.

J'ai observé qu'un binôme était dans la co-construction du programme comme cela avait été fait dans la séance 2 en débranché. Les deux élèves se mettent d'accord ensemble sur le code à écrire. Elles inscrivent chacune à leur tour les lignes de codes dans la feuille de programmation.

Verbatim de deux élèves réalisant l'activité 1 :

Élève 1 : « Donc là c'est avance. »

Élève 2 : « Je pense... On peut toujours essayer. »

Élève 1 : « Oui. Et heuu... »

Élève 2 : « Jaune ! »

Élève 1 : « Jaune, oui c'est ça... »

Élève 2 : « Ah, c'est à moi de le faire » (prenant les commandes et plaçant la commande « jaune »)

Élève 1 : « Ah bah oui. »

Élève 2 : « Et ... »

Élève 1 : « Et dessiner, là dessiner » (en montrant sur l'écran)

Élève 2 : « Dessiner... vas-y fais le parce que si... je l'ai fais la dernière fois. »

Élève 1 : « Dessiner... Ouais ça marche ! A moi de faire... »

Élève 2 : « Bah du coup là c'est à moi de faire « avancer » »

Élève 1 : « Ah oui et moi je fais les couleurs. »

Elles choisissent de conserver cette modalité de travail où tour à tour elles ajoutent des ordres pour construire un programme de déplacement pour les activités suivantes. Lorsqu'elles sont à l'activité 7, ces deux élèves changent de stratégies. Elles pensent qu'elles seront plus efficaces si l'une dit et l'autre fait. Elles se répartissent les rôles ainsi et décident qu'elles échangeront à chaque nouvelle activité. Une élève réfléchit au code qu'il faut inscrire et l'autre prend en charge la rédaction du code sous la dictée de sa camarade. Cela leur permet de retirer une part de charge mentale puisqu'elles n'ont plus les deux rôles à assumer en même temps. J'ai observé un autre binôme dans lequel les deux élèves s'étaient partagés les rôles de la même façon. L'un était le programmeur, il devait réfléchir aux ordres qu'ils allaient devoir donner à la flèche pour tracer le motif demandé, l'autre étant l'exécutant. Il ajoutait les ordres sur la feuille de code au fur et à mesure que le programmeur les lui dictaient. Ils ne vérifiaient leur tracé qu'après avoir entré toutes les lignes de code.

Dans la classe de CM2, j'ai constaté que les élèves préféraient se séparer les activités. Dans les binômes il y avait un élève qui réalisait l'activité 1 et lorsqu'il avait fini, c'était à l'autre élève de faire l'activité suivante. Pendant qu'un des membres du binôme raisonnait sur le problème qui lui était posé, le second pouvait aider en donnant son avis ou en faisant des suggestions. Dans 3 binômes, j'ai observé que le deuxième élève passif se désintéressait de la tâche et laissait son

binôme travailler seul pendant qu'il allait voir où en étaient ses camarades de classe et comment ils se débrouillaient pour résoudre leur activité en cours.

Comme il s'agissait d'une première expérience pour les CM2 sur des activités de programmation, ils avaient le besoin de réfléchir seul le temps de comprendre le fonctionnement de l'environnement et le langage de codage qu'ils devaient utiliser. Les CM1-CM2 avaient pu avoir ce temps d'apprentissage en individuel lors de la séance 2 et 3 ce qui leur avait permis de construire leurs représentations. C'est en partie grâce à ce moment d'assimilation en individuel que les temps de travaux de groupe pouvaient être plus pertinents car les élèves pouvaient confronter leurs visions.

Le fait d'avoir pratiqué la programmation en débranché a permis de confronter les élèves à des situations dans lesquelles il était nécessaire de se répartir les rôles pour travailler efficacement. Cela avait été clairement verbalisé lors de la séance 4 qu'il était plus efficace de travailler en coopération avec les autres membres du groupe pour raisonner ensemble sur les situations proposées. Ils ont pu appliquer cette stratégie lors de la séance de réinvestissement sur un logiciel de programmation et étaient davantage dans un travail de binôme car les deux élèves étaient acteurs dans le travail. A l'inverse, les élèves de CM2 n'ayant pas été confrontés à ces situations, il n'ont pas tous perçu de nécessité à réfléchir ensemble aux situations présentées.

4. 1. 4. Le conflit socio-cognitif

Lorsqu'ils sont amenés à faire un travail de groupe, les élèves peuvent faire face à des situations de conflit socio-cognitifs où des points de vues, des procédures s'opposent. Comme les élèves n'avaient qu'un ordinateur par binôme, ils devaient coopérer pour résoudre les situations présentées. J'ai pu observer deux élèves de la classe de CM1-CM2 s'opposer dans leur démarche de résolution. Chacune essayant de défendre sa position à l'autre pour finalement trouver un consensus qui leur a permis de résoudre l'activité. Elles souhaitaient utiliser la boucle pour résoudre l'activité 9 mais ne parvenaient pas à trouver comment l'utiliser efficacement. L'une des élèves voulait persévérer dans l'usage de la boucle et s'en servir pour résoudre la situation alors que l'autre voulait abandonner la boucle et inscrire chaque ligne sans faire de répétition. Après quelques minutes de débat, la première a pu tester un usage de la boucle mais ne parvenant toujours pas à s'en servir avec efficacité dans le cadre de cette activité, elle a accepté la proposition de sa camarade à savoir inscrire chaque ligne sans user de la boucle de répétition. Ce type d'interactions amène les élèves à

devoir argumenter sur leur position, cela montre alors qu'ils sont dans le registre d'interprétation et qu'elles se dirigent vers le registre de modification. Ces deux élèves ont compris les différentes commandes qu'elles ont à disposition et sont capables de débattre d'une stratégie à adopter pour résoudre le problème efficacement.

Ce type d'interactions n'a pas été observé dans la classe de CM2. Les binômes du groupe de CM2 s'étaient répartis le travail de façon à ce qu'un élève soit programmeur et l'autre observateur. Les interactions étaient limitées car l'observateur restait passif et n'était pas dans l'interprétation du programme rédigé par son camarade. Il y a eu très peu de co-construction dans les binômes observés de la classe de CM2.

Je pense que l'écart observé entre les deux groupes est également dû au fait que les élèves de la classe de CM1-CM2 avaient été amenés à interpréter et raisonner sur des séquences de programmation dans les séances précédentes. Lorsqu'ils devaient appairer une scène avec un programme, il leur avait fallu anticiper un résultat et interpréter un programme. Ils ont pu réinvestir cela lorsqu'ils étaient confrontés aux activités sur le logiciel de programmation. Ils étaient capables de se faire une idée de la procédure pour parvenir au résultat et pouvaient entrer en conflit socio-cognitif si le partenaire n'avait pas la même idée alors que le groupe de CM2 n'a pas été entraîné à anticiper le résultat d'un programme. Ils démarraient les activités sans se faire une idée du programme qu'ils devaient écrire.

4. 1. 5. Gestion du temps

En étudiant les traces recueillies grâce au logiciel de Pixel'Art, j'ai pu constater le temps que mettaient les binômes de la classe de CM2 pour résoudre les situations proposées.

J'ai comparé les résultats des deux binômes les plus performants dans chacun des groupes. Leur méthode de travail était assez similaire : ils inscrivaient l'intégralité du programme avant de le dessiner pour constater leur réussite ou leurs erreurs. J'ai comparé le temps nécessaire à chacun de ces deux binômes pour résoudre les activités et j'ai cherché à expliquer les écarts observés (voir figure 11).

Pour le binôme de CM2 étant allé le plus loin dans les activités proposées, il leur a fallu environ 6 minutes 50 secondes puis 6 minutes pour résoudre chacune des deux premières situations permettant de prendre en main le logiciel. Pour le binôme le plus performant de la classe de CM1-CM2, l'activité 1 a été résolue en 1 minute puis 2 minutes 20 secondes pour la deuxième activité.

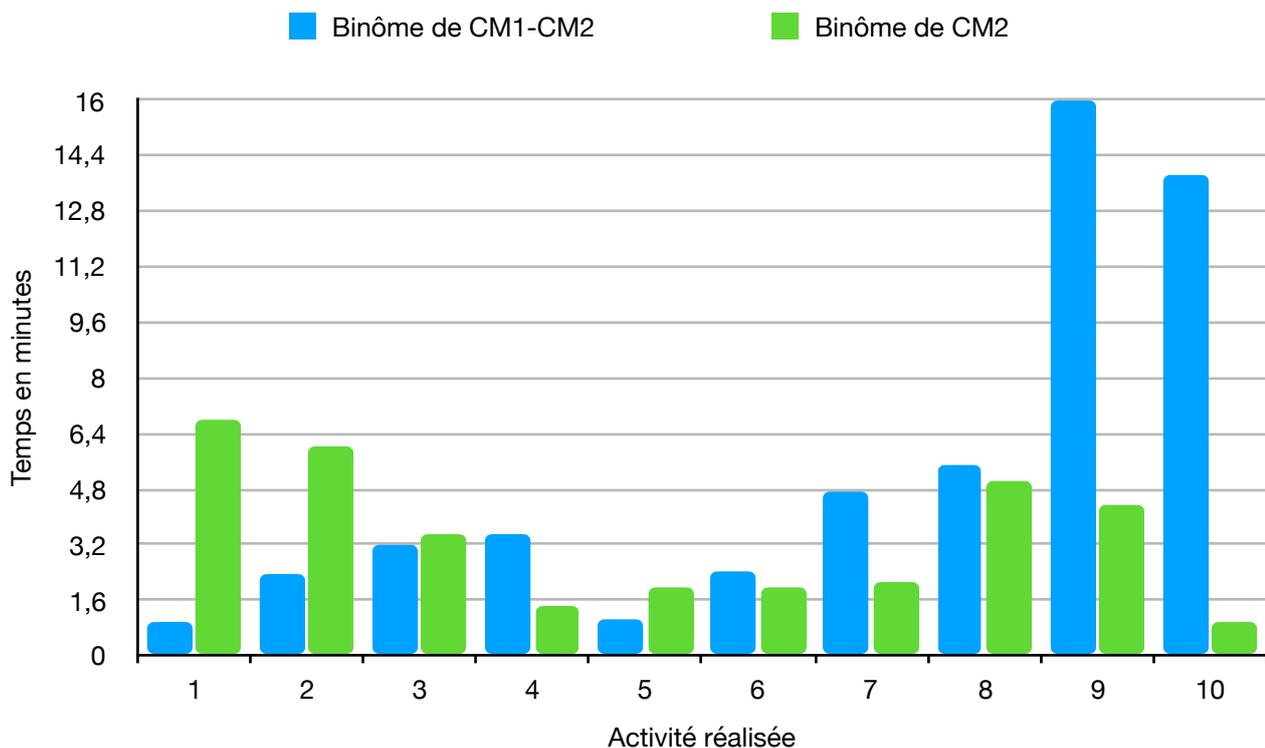


figure 11. Temps en minutes nécessaire pour résoudre chaque activité pour deux binômes observés

On peut en déduire que la prise en main du logiciel était facilitée pour les binômes ayant travaillé sur des activités de programmation en débranché et que cela a été un obstacle pour le binôme de CM2. Il ont eu besoin de plusieurs minutes pour comprendre le langage du codage et comment l'utiliser.

Par la suite, le binôme de CM2 mettait beaucoup moins de temps pour résoudre les situations proposées. Ils savaient utiliser les commandes de base et résolvaient rapidement les différentes activités. Les deux binômes comparés ont eu besoin du même temps (environ 3 minutes) pour résoudre l'activité 3.

On constate ensuite que le binôme de la classe de CM1-CM2 a eu besoin de 3 minutes 32 secondes pour résoudre l'activité 4 contre 1 minute 26 secondes pour le binôme de CM2. Les deux élèves de CM1-CM2 ont en effet voulu optimiser leur travail en utilisant pour la première fois la boucle afin de répéter une séquence d'ordres. Quelques manipulations étaient nécessaires pour bien s'imprégner de son utilisation et pour pouvoir s'en servir efficacement dans l'activité. Mais elles ont pu directement réinvestir l'utilisation de la boucle de répétition dans l'activité 5 et ainsi la résoudre en 1 minute. Le binôme de CM2 observé a eu besoin de 2 minutes pour résoudre cette activité en faisant figurer tous les ordres sur leur feuille de programme.

Le temps passé sur l'activité 6 est similaire entre les deux binômes et est d'environ 2 minutes. Les élèves de CM1-CM2 ont également utilisé la boucle pour résoudre cette situation et il leur a fallu comprendre dans quelle ordre placer les différentes commandes pour que l'utilisation de

la boucle soit correcte. Le temps nécessaire pour résoudre l'activité 8 est également similaire dans les deux binômes (environ 5 minutes) mais on constate cette fois là que le binôme de CM1-CM2 a utilisé la boucle de répétition à 4 reprises.

Lors de l'activité 7, le binôme de CM2 n'a eu besoin que de 2 minutes et 10 secondes pour écrire un programme correct alors que les élèves de la classe de CM1-CM2 ont eu besoin de 4 minutes 44 secondes. Cela s'explique par le fait qu'elles ont à nouveau voulu utiliser la boucle. Cependant après deux essais elles ne parviennent pas à l'utiliser de façon efficace dans cette situation et préfère renoncer en se contentant d'inscrire sur leur programme la succession d'ordre sans utiliser la commande de répétition. Cette situation se reproduit lors de la résolution de l'activité 9. Les CM2 ont eu besoin de 4 minutes et 23 secondes en continuant d'utiliser uniquement les ordres de base sans recourir à la répétition alors que le binôme de CM1-CM2 a eu besoin de 16 minutes. Les deux élèves ont à nouveau voulu recourir à l'usage de la boucle et ont fait plusieurs tentatives d'utilisation. Après avoir fait une première tentative de l'usage de la boucle, elles constatent qu'elles n'obtiennent pas le résultat attendu (voir figure 12).

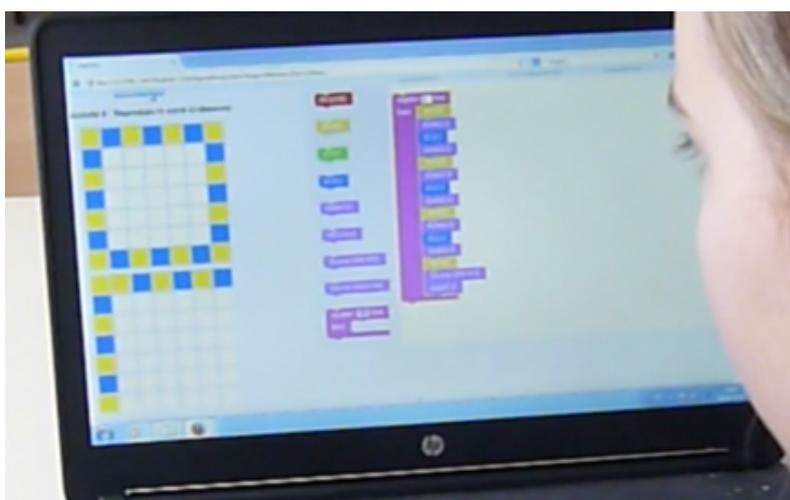


figure 12. Usage de la boucle dans le binôme de CM1-CM2

Elles cherchent à modifier les ordres inscrits dans leur programme à plusieurs reprises mais ne parviennent pas à identifier la source de leur erreur. Elles finissent par abandonner l'usage de la boucle pour inscrire tous les ordres les uns à la suite des autres. Malgré qu'elles n'aient pas réussi à se servir efficacement de la commande de répétition dans deux situations, on constate tout de même que ces élèves sont dans le registre de modification. Elles ont compris l'usage des différentes commandes et y ont recours de façon pertinente. Elles ont cependant été confrontées à quelques obstacles qu'elles n'ont pas réussi à surmonter.

Par la comparaison de la gestion du temps, on constate que le binôme de CM2 visait la rapidité d'exécution en utilisant les commandes découvertes dans les premières activités sans s'interroger sur la commande de répétition. Le binôme de CM1-CM2 a mis plus de temps à résoudre certaines situations mais cherchait à écrire un programme qui utilisera de façon efficace les différentes commandes mises à disposition par le logiciel. Cet écart s'explique également par les activités de programmation en débranché réalisées lors des séances 2 et 3 dans lesquels les élèves étaient amenés à raisonner pour élaborer le programme nécessitant l'usage d'un minimum de commandes. Les élèves de CM1-CM2 ont été entraînés à chercher une stratégie de résolution non pas basée sur la rapidité de résolution mais sur la pertinence du programme et des commandes utilisées.

4. 2. Découverte du logiciel Pixel'Art

4. 2. 1. Lecture et compréhension des commandes

Les élèves ayant déjà expérimenté des tâches de programmation en débranché avaient une première connaissance du système d'écriture en langage de code. Ils savaient qu'un code se lit de haut en bas. Pour la première activité, la flèche se situe en bas du quadrillage et il faut la faire monter et colorier les cases se trouvant sur son chemin. Cette activité n'a pas posé de problème aux élèves de la classe de CM1-CM2 qui ont rapidement résolu la situation. Ils étaient déjà dans le registre d'interprétation au début de l'activité car ils étaient en capacité d'analyser la tâche demandée ainsi que les commandes qu'il était nécessaire d'utiliser pour réussir cette activité.

Dans la classe de CM2 n'ayant pas travaillé sur les activités de programmation en débranché, la compréhension du sens de lecture a été un premier obstacle. Sur la figure 13, on peut voir que l'élève met dans les lignes de code les couleurs de la figure à tracer mais en les indiquant dans le même ordre qu'elle voit de haut en bas. Elle inscrit également toute les lignes de couleur sans utiliser la commande « avance » et elle n'est pas encore dans une situation de décentration en ne prenant pas en compte la flèche qu'il faut déplacer. Ce n'est qu'une fois qu'elle a indiqué les 4 ordres que l'on voit dans la figure 13 qu'elle va cocher tout de suite « pas à pas » et « voir la trace » avant de lancer le dessin.

Verbatim d'élèves de la classe de CM2 :

Élève A : « T'as vu ça n'en met qu'une » (en parlant des couleurs)

(l'élève B ayant entré les lignes de codes clique à plusieurs endroits et fini par cocher la case « pas à pas » avant de cliquer à nouveau sur « dessiner ».)

Élève A : « Oh ! T'as vu c'est trop bizarre quand tu mets une couleur ça t'en met qu'une ! » (en parlant des couleurs)

(...)

Elève A : « Et comment on fait pour aller en haut ? » (l'élève A observe l'écran d'un autre binôme)

Elève A : « Ah je sais ! Faut mettre « avance » ! Faut mettre « avance » ! Tu mets « avance ». » (en montrant avec sa main)

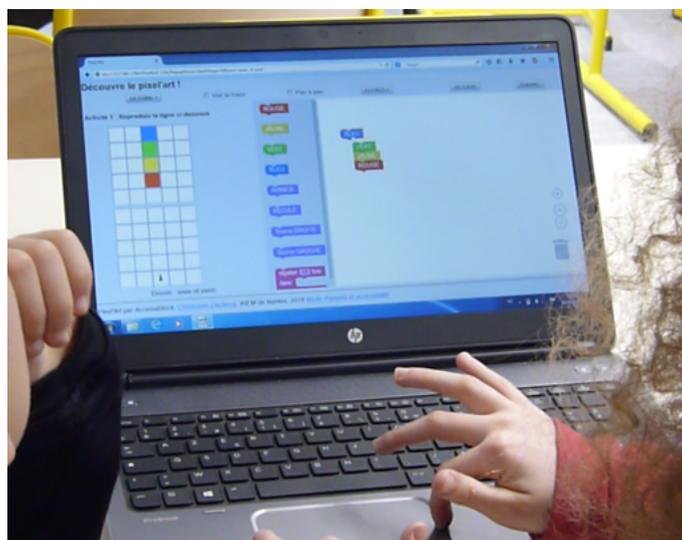


figure 13. Elèves de CM2 résolvant l'activité 1

Dans cet échange, nous comprenons que les élèves sont en train de résoudre le problème par tâtonnement en testant plusieurs boutons pour voir si le résultat sera différent. Ils sont dans le registre de participation tel que le décrit O. Grugier car ils explorent les fonctionnalités offertes par le logiciel. Grâce à l'observation du travail d'un autre binôme, l'élève A commence à percevoir le sens des différentes commandes à disposition. Le binôme comprend par la suite qu'il faut déplacer une flèche pour pouvoir colorer les différentes cases. Ils rentrent alors dans le registre d'interprétation.

La deuxième difficulté rencontrée par des élèves de CM2 n'ayant pas réalisé de tâches de programmation en débranché est dans l'organisation de la feuille de code. Ils vont sélectionner les codes qu'ils souhaitent utiliser et les placer dans la feuille de code les uns accrochés à la suite des autres. En testant leur programme et en constatant qu'il y a des choses qui ne fonctionnent pas, ils

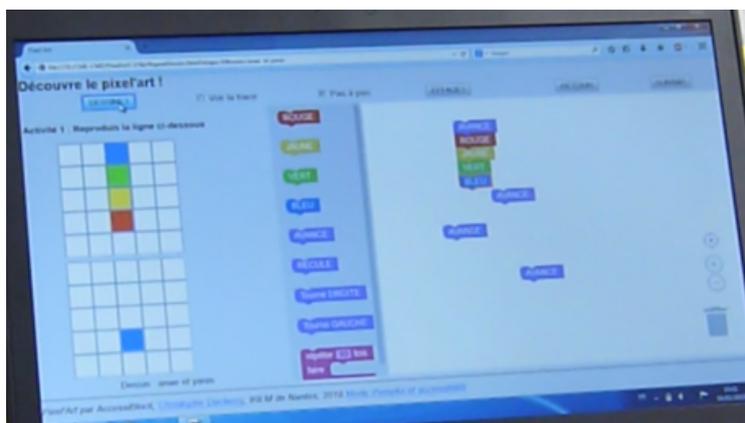


figure 14. Environnement de travail du logiciel Pixel'Art

vont simplement détacher des codes mais en les gardant sur le côté de leur feuille de code (voir figure 14).

Cela occasionnera des difficultés de compréhension dans le programme qu'exécute la flèche lorsqu'ils font l'action de démarrer. Ils ne perçoivent pas que tous les ordres inscrits dans la feuille de code sont exécutés même s'ils ne sont plus attachés les uns aux autres.

Cette difficulté n'a pas été rencontrée dans le groupe de CM1-CM2 ayant travaillé la programmation en débranché. Les élèves prenaient toujours soin d'attacher les ordres et de le glisser hors de la feuille lorsqu'ils voulaient le supprimer. Je pense que cela est dû à la fréquentation de programmes lors des séances précédentes dans lesquels les lignes de codes étaient bien attachés entre elles dans un bloc de code.

4. 2. 1. Essais du programme

J'ai constaté une évolution dans la procédure de test des programmes écrits par les élèves. J'ai observé que dans un binôme de la classe ayant travaillé sur des activités débranchées, les élèves font le choix de vérifier régulièrement leur production. Ainsi, ils s'assurent d'avoir choisi les bons ordres et d'avoir la bonne interprétation des ordres inscrits. Cela leur permet de se conforter dans leur compréhension des ordres disponibles et de pouvoir faire le lien avec le langage de codage qui avait été travaillé dans les séances précédentes.

Verbatim de deux élèves réalisant l'activité 1 :

Élève 2 : « Oh peut-être qu'on devrait mettre tout d'affilé et après on dirait dessine. »

Élève 1 : « Heu ouais on peut essayer mais ça va peut-être pas marcher. »

Élève 2 : « Bon heu tu mets la couleur. »

Élève 1 : « Ok. »

Pour la suite ces deux élèves tentent d'écrire tout le programme avant de le tester. Ainsi elles travaillent d'avantage l'anticipation en s'imaginant ce qui sera réalisé. L'élève qui avait proposé de travailler ainsi va s'en faire la réflexion un peu plus tard « C'est beaucoup plus pratique de tout calculer et après dire dessiner. ».

À l'image de la stratégie initiale des deux élèves décrite précédemment, la plupart des binômes de la classe de CM1-CM2 faisaient le choix dans un premier temps de tester à chaque ajout de ligne de code pour s'assurer de leur compréhension avant de faire évoluer leur stratégie pour inscrire plusieurs lignes de code avant de tester le programme. Ils étaient capables d'anticiper l'enchaînement des instructions qu'ils inscrivaient pour voir où ils en étaient dans la rédaction de leur programme.

Dans le groupe des CM2, les élèves indiquaient plusieurs lignes de code, avant de tester leur programme. Ils ont davantage recouru à l'utilisation de la commande « pas à pas » pour avoir un retour instrumental sur le programme qu'ils venaient d'inscrire ce qui était moins fréquent dans le l'autre groupe classe. Les élèves de CM1-CM2 étaient capables de visualiser l'avancée de la flèche sans avoir nécessairement besoin du retour instrumental. J'ai constaté cela lorsqu'un binôme de CM1-CM2 était bloqué sur une activité. L'élève a pu m'indiquer sur le quadrillage où il prévoyait que la flèche arriverait après l'exécution du programme qu'il avait écrit. Il était capable d'anticiper le résultat du programme inscrit sans l'avoir testé auparavant. Cela peut s'expliquer par le travail en activité débranché sur l'anticipation et la lecture des programmes lors des séances 4 et 5. Les élèves s'étaient entraînés à se représenter le déplacement d'un personnage sur un quadrillage.

4. 3. Usages du langage du codage

4. 3. 1. Les commandes « tourner à gauche » ; « tourner à droite »

Lors de l'activité 2, les élèves sont confrontés à la réalisation d'un dessin dans lequel ils devront faire pivoter la flèche sur la droite. J'ai constaté que, de façon générale, les élèves des deux classes étaient dans un premier temps surpris de constater que lorsqu'ils inscrivaient la ligne de

code « tourne à droite » la flèche n'avancait pas d'une case. La figure qu'ils traçaient ne correspondait pas à leurs attentes (voir figure 15)

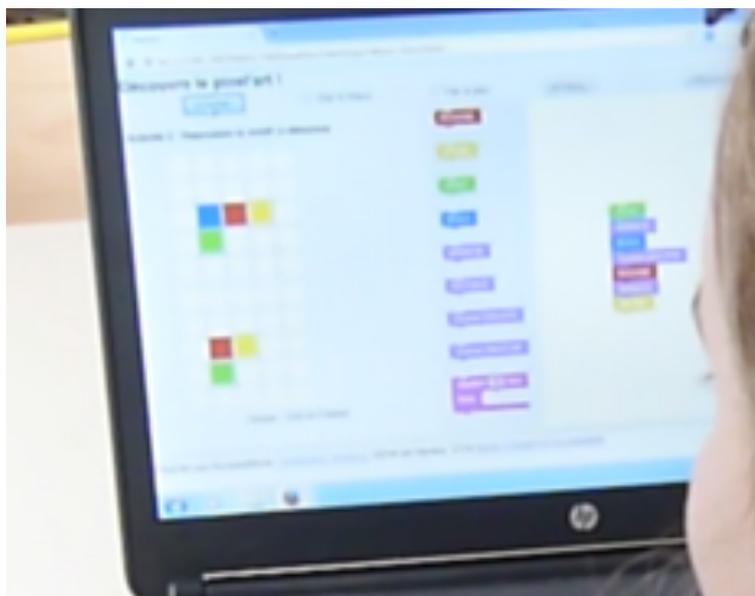


figure 15. Premier usage de la commande « tourner à droite »

Certains binômes sont parvenus à surmonter cette difficulté en testant leur code au fur et à mesure afin de voir à quel moment il y avait un problème dans le code qu'ils avaient inscrit. Ils s'apercevaient de ce problème de flèche qui n'avancait pas et se rendaient compte qu'en ajoutant la commande « avance » le dessin qu'ils voulaient faire apparaissait. J'ai tout de même dû intervenir auprès de certains binômes de la classe de CM1-CM2 en leur posant une question : « Si je dis la commande 'tourne à droite', que fait le personnage sur le quadrillage ? ». En leur demandant cela, les élèves comprenaient alors qu'on ne donnait pas l'instruction d'avancer mais seulement de tourner et que c'était pour cela que la flèche n'avancait pas. Ils étaient capables d'appliquer cela à leur code et de résoudre le problème qui s'était posé. Le simple fait de leur poser la question leur permettait de comprendre où se situait leur erreur. Ils avaient déjà été confronté à cela lors de la séance 2 avec la situation du robot idiot au cours de laquelle il avait été mis en évidence que lorsqu'on donnait la consigne de « tourner » le robot ne changeait pas de case.

Dans le groupe de CM2, certains élèves ont été confrontés à la même difficulté mais le retour instrumental du logiciel a été nécessaire pour qu'ils se rendent compte que la flèche n'avancait pas lorsqu'elle tournait à droite. Les élèves testaient leur programme à chaque fois qu'ils ajoutaient une ligne de code pour comprendre où se situait l'erreur, ils se rendaient compte alors que la flèche restait sur la case lorsqu'ils donnaient la consigne « tourne DROITE ». D'autres binômes ont eu besoin de recourir au dessin pas à pas pour voir qu'il manquait la commande « avance »

après l'ordre de tourner à droite pour que le motif soit correct. Les élèves du groupe de CM2 adoptaient une posture de programmeur pas à pas telle que Christophe Declercq et Françoise Tort la décrivent pour résoudre les situations qui faisaient obstacle à leur méthode de résolution.

4. 3. 2. La boucle de répétition

L'usage de la boucle avec la notion de répétition d'une commande un certain nombre de fois est induite dans l'activité 4 (voir figure 16).

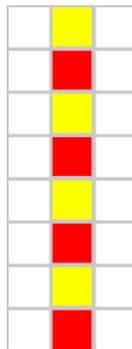


figure 16. Motif à reproduire pour l'activité 4.

Dans un binôme observé de la classe de CM1-CM2, une élève commence par indiquer tous les codes : « Rouge » ; « Avance » ; « Jaune » ; « Avance » ; « Rouge » ; « Avance » ; « Jaune » ; « Avance ». C'est après avoir ajouté ce dernier ordre que l'une des deux élèves propose l'utilisation de la commande « Répète ». Il y a alors une confrontation et un désaccord entre les deux élèves.

Verbatim de deux élèves réalisant l'activité 4 :

Élève 1 : « Ah mais non ! Je sais ce qu'on pourrait faire ! »

Élève 2 : « Non, non non ! »

Élève 1 : « Si si ! Bah là on a juste à faire ça... » (en sélectionnant le bloc de répétition)

Élève 2 : « Bah non c'est pas dix fois. »

Élève 1 : « Si là on va enlever on va pas mettre 10 ... Là on en est à 'rouge' 'avance'... » (en sélectionnant le bloc de répétition)

Élève 2 : (après avoir compté les blocs sur l'écran) « Quatre. Et du coup efface jusque là. » (en montrant sur l'écran après le second ordre « avance » entré dans le programme et en prenant les commandes) « Et du coup je décale ça » (l'élève place à côté du programme écrit l'ordre de répétition).

Après avoir réduit le programme aux quatre ordres qu'elle souhaitait conserver, l'élève 2 place l'ordre de répétition après les quatre ordres déjà présents dans leur programme puis elle teste le programme (voir figure 17).

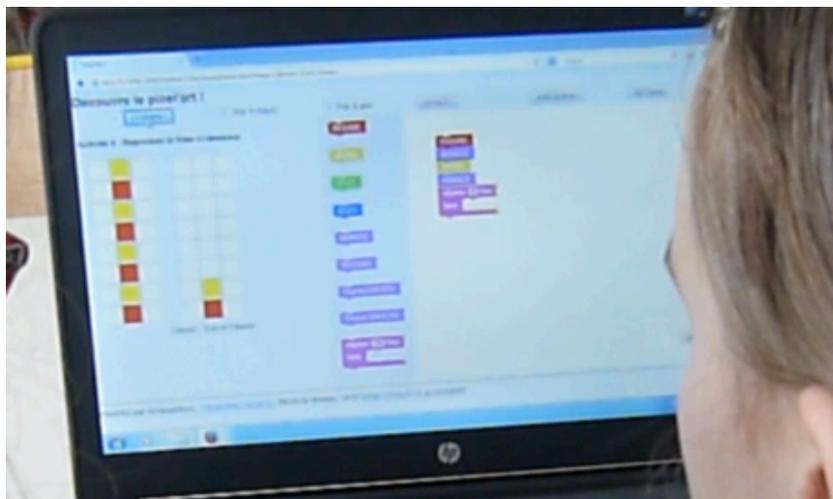


figure 17. Premier usage de la boucle de répétition dans un binôme de CM1-CM2

Dans un premier temps, les élèves sont dans l'incompréhension. L'élève 2 en conclut même que cette commande ne fonctionne pas.

Verbatim de deux élèves réalisant l'activité 4:

Élève 2 : (en supprimant le bloc de répétition) « Bah non, du coup ça a pas fonctionné. C'est bien ce que je pensais. »

L'élève 1 reprend ensuite les commandes, elle place la commande de répétition et fait glisser les ordres déjà inscrits dans la répétition. L'élève 1 modifie la valeur de répétition à 4 mais il y a de nouveau un désaccord entre les deux élèves. L'élève 2 pense qu'il faut réduire à 3 en pensant que le programme va appliquer une fois les ordres qu'elles ont inscrits puis les répéter 3 fois par la suite, elle pense donc qu'indiquer 3 fois suffirait. L'élève 1 soutient que non et ne permet pas à l'élève 1 de vérifier sa théorie, elle valide le programme et le motif s'affiche. L'élève 2 reste sur son idée qu'il aurait fallut réduire à 3. Dans cette manipulation, on voit que les deux élèves ont compris quand l'usage de la boucle était utilisable pour créer un programme plus court mais on constate que dans la mise en oeuvre, elles ont eu besoin de procéder par essai-erreur pour trouver la bonne utilisation à en faire. Dans l'application de la boucle, il subsiste un malentendu chez l'élève 1. De

façon générale, on constate que ces deux élèves ont su y recourir au moment approprié et ont rapidement su s'en servir efficacement.

Dans la classe de CM1-CM2, parmi les 11 binômes observés, 7 ont eu recours à l'usage de la boucle. Parmi les 4 binômes observés n'ayant pas utilisé la boucle, deux sont composés d'élèves d'élèves de CM1. Ils avaient moins d'aisance dans l'usage de la boucle et utilisaient en priorité les commandes de base avec la volonté de rester dans leur zone de confort.

J'ai observé que certains binômes ne commençaient à recourir à l'usage de la boucle qu'à partir de l'activité 8. Leur objectif n'étant pas d'éviter d'avoir à répéter plusieurs fois une même succession d'ordre mais de gagner de la place sur la feuille de code en réduisant la quantité d'ordres à y inscrire comme on le perçoit dans un échange entre deux élèves :

Verbatim de deux élèves réalisant l'activité 8 :

Élève 3 : « Ça va être simple ça ! »

Élève 4 : « Mais on n'a pas la place... »

Élève 3 (en montrant le bloc de répétition) : « Hé mais t'as qu'à faire ça ! Répète je sais pas combien de fois... »

(ils placent l'ordre à la suite de leur programme et constatent que ça ne fonctionne pas, ils suppriment l'ordre et inscrivent tous les ordres les uns à la suite des autres)

(après avoir renseigné plusieurs ordres dans leur feuille de code)

Élève 4 : « Maîtresse ! Mais en fait on aura pas la place de terminer alors qu'on est rendu que là... »

(en montrant sur le quadrillage où ils prévoient d'arriver après exécution de leur programme)

PE : « Ah, alors, comment est-ce qu'on peut faire ? »

Élève 4 : « On peut mettre à côté ? »

Élève 3 : « Bah moi je dis qu'on fait ça ! » (l'élève coche le 'voir la trace') « Ah bah non. »

PE : « Qu'est-ce qu'on peut faire alors ? »

Élève 4 : « Bah déjà on peut mettre dessine... »

PE : « Dans les ordres, qu'est-ce qui te permettrait d'éviter d'avoir autant d'ordres ? »

Élève 4 : « 'Pas à pas' ! » (l'élève teste) « Non... »

Élève 3 : « Non ! Répète 5 fois mais ça a pas marché l'autre fois. On a mis ça a pas marché. Parce que ça a mis que un carreau. »

(Les élèves continuent de tâtonner en testant différents ordres. Après plusieurs manipulations les élèves comprennent comment utiliser la boucle et glissent les commandes à l'intérieur.)

Élève 3 : « Mais du coup on a la place maintenant ! »

Les élèves économisent ainsi de la place sur leur feuille de programmation, le code qu'ils inscrivent est moins long et est donc plus lisible pour eux.

L'usage de la boucle était moins systématique dans la classe de CM2, seulement 5 binômes ont recouru à son utilisation. Certains élèves avaient compris son fonctionnement et essayaient d'amener d'autres binômes à s'en servir. Par exemple, un binôme observé dans la classe de CM2 n'a pas eu recours à l'usage de la boucle. Ils ont indiqués tous les ordres successivement (voir figure 18) Lorsqu'ils s'apprêtent à lancer leur programme, une camarade vient leur faire remarquer qu'ils peuvent utiliser la commande de la répétition.

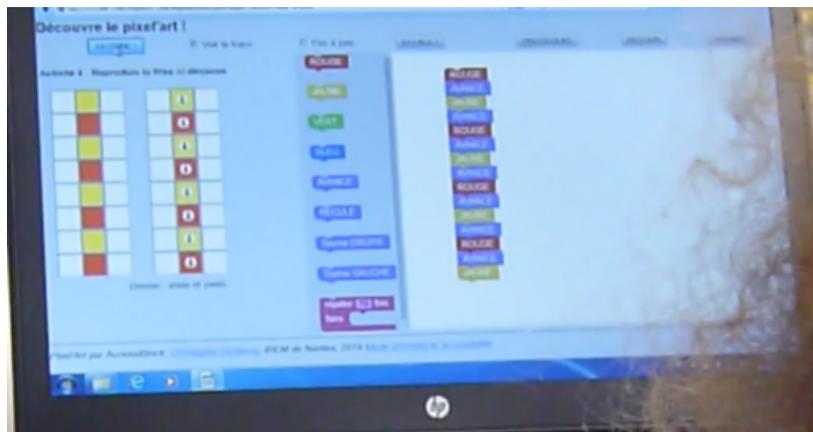


figure 18. Résolution de l'activité 4 par le binôme de CM2

Verbatim d'une élève montrant l'usage de la commande répéter à un autre binôme :

Élève C (ne faisant pas parti du binôme en pointant du doigt la commande de répétition) : « Vous savez que vous pouvez faire ça ? Et que vous pouvez écrire autre chose à la place du 10. »

Les deux élèves du binôme ne répondent pas et l'élève A finit d'écrire le programme sans prendre en compte cette commande conseillée par l'Élève C. Lorsqu'ils commencent l'activité suivante, l'élève C déplace le curseur de la souris pour le positionner sur la commande répéter et conseille à nouveau à ses camarades de s'en servir. L'élève B commence ensuite à écrire le programme (voir figure 19)

L'activité 5 est similaire à la 4 mais en incluant une variation de 3 couleurs au lieu de deux. L'élève B procède comme précédemment en inscrivant les ordres sans utiliser la boucle.

L'élève C intervient encore à deux reprises pour conseiller l'usage de la boucle à ses camarades mais ceux-ci conservent leur stratégie initiale et inscrivent tous les ordres sans prendre en compte les recommandations de l'élève C. On constate même qu'ils ne lui répondent pas lorsqu'elle leur

conseille l'utilisation de cette commande. On peut supposer que, ne sachant pas s'en servir, ils préfèrent continuer à utiliser les commandes qu'ils maîtrisent sans prendre de risques. Ils choisissent de rester dans leur zone de confort en utilisant les commandes qu'ils comprennent sans prendre le risque d'en essayer une dont l'utilisation n'est pas maîtrisée. Ils ne sont pas dans l'utilisation optimale du logiciel et restent dans le registre d'interprétation.

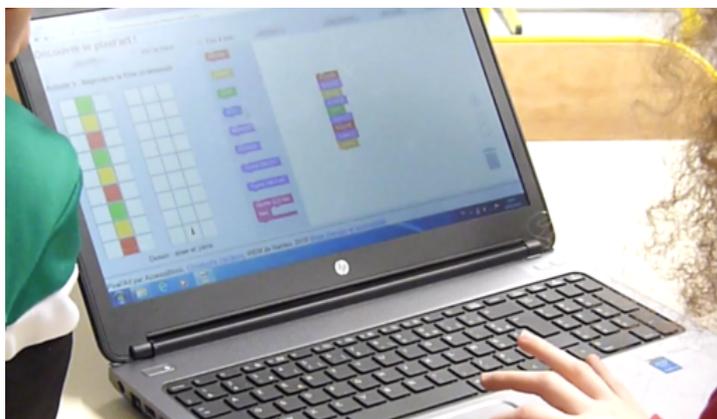


figure 19. Résolution de l'activité 5 par le binôme de CM2

Seulement 5 binômes ont essayé de recourir à l'usage de la boucle et de la répétition dans la classe de CM2, cela représente 35,7% du groupe. Même le binôme qui a été conseillé par une de leur camarade n'a pas du tout recouru à l'usage de la boucle par la suite pour résoudre les situations proposés. En revanche, 7 binômes sur les 11 observés soit 63,6% des élèves de la classe de CM1-CM2 y ont recouru même si certains ont eu des difficultés à l'utiliser. Il leur a fallu procéder par tâtonnement pour bien percevoir comment l'intégrer dans leur feuille de code. Je pense que la séance débranchée faisant travailler la mise en lien entre des scènes et des feuilles de code incluant des boucles a permis aux élèves de la classe de CM1-CM2 de se familiariser avec l'utilisation de la boucle en programmation. Cela leur a également permis de percevoir son utilité pour réduire la quantité de lignes de codes à intégrer à leur programme. On le constate dans un échange entre deux élèves pendant l'activité 8. La pratique d'activité débranchée a donc eu un impact sur la fréquence de l'utilisation de la boucle chez les élèves. J'avais supposé que les élèves réinvestiraient facilement les éléments du codage qu'ils avaient fréquentés lors des activités en débranché. S'ils n'ont pas montré de difficulté à utiliser les commandes de base du logiciel, une résistance était toujours présente lorsqu'il s'agissait d'utiliser une commande plus complexe comme la boucle de répétition. Cependant davantage de binômes ont tenté d'y avoir recours dans le groupe de CM1-CM2 comparé à celui des CM2. Dans les séances en débranché, les élèves devaient uniquement interpréter l'usage de la boucle de répétition mais ils n'ont pas été mis en situation d'utilisation. Pour que les élèves

s'en servent avec davantage d'aisance, il aurait été nécessaire de faire plusieurs séances en débranché avec pour objectif de familiariser les élèves à la compréhension et à l'usage de la boucle.

4. 4. Stratégies observées face à l'erreur

4. 4. 1. Réinitialisation du programme

Lorsqu'un binôme testait son programme et qu'il constatait que le résultat n'était pas celui escompté, il supprimait le programme et recommençait depuis le début. Parfois même, lorsque c'était un élève qui avait rédigé seul le programme testé, il laissait la main à son camarade qui reprenait tout depuis le début. J'ai observé que cette stratégie a beaucoup été utilisée dans le groupe de CM2 mais je ne l'ai pas observée dans la classe de CM1-CM2. Les élèves agissant ainsi étaient soit dans le registre d'interprétation : ils avaient compris le fonctionnement du logiciel mais ne parvenait pas à l'utiliser efficacement pour raisonner sur leur proposition afin de déceler l'erreur, soit ils étaient dans le registre précédent, celui de participation. Ils testaient de façon plus aléatoire les commandes et reprenaient à zéro après chaque essai pour essayer de trouver un programme qui exécuterait les commandes attendues.

4. 4. 2. Programmeur pas à pas

Dans les deux groupes, j'ai observé que lorsqu'ils se trompaient, certains élèves repartaient depuis le début de leur programme et testaient ordre par ordre afin de voir à quel moment l'erreur était commise. Ils avaient une utilisation pas à pas du logiciel pour déceler avec plus de précision l'erreur qu'ils avaient fait. Les élèves utilisant cette stratégie terminaient souvent l'activité en cours en restant dans la posture du programmeur pas à pas. Cela aurait peut-être pu être évité si les fonctionnalités « voir la trace » et « pas à pas » avaient été présentées au début de la séance. Ainsi les élèves auraient pu se saisir en autonomie du retour instrumental que propose directement le logiciel. Je suis intervenue auprès de certains binômes pour leur présenter cette commande et son utilité. 4 binômes y ont eu recours dans la classe de CM1-CM2 soit environ 36,4% des élèves contre 6 binômes dans la classe de CM2 ce qui représente 42,9% du groupe. Cette différence s'explique que les élèves de CM1-CM2 avaient appris dans les séances en débranché à anticiper les déplacements d'un personnage en lisant un programme. L'utilisation de la commande pas à pas

n'étaient pas une nécessité pour eux puisque les binômes parvenaient à se représenter eux-même le programme qu'ils avaient rédigé pour déceler une erreur.

4. 4. 3. Changement de programmeur

Cette stratégie a été principalement observée dans le groupe de CM2. Lorsqu'une erreur était constatée dans l'exécution du programme, les deux élèves s'échangeaient les rôles. Celui qui venait d'écrire le programme laissait la main à son camarade. Ce dernier essayait de relire le programme écrit et de visualiser les actions qui seront exécutés par la flèche.

4. 4. 4. Demande de solution à un tiers

Dans le groupe de CM2, lorsqu'un binôme ne parvenait pas au résultat, j'ai également observé qu'ils faisaient rapidement appel à un tiers : moi-même ou à un camarade ayant déjà résolu l'activité. Les élèves demandaient le résultat sans chercher à raisonner sur ce qu'ils avaient produit et où pouvait se trouver leur erreur. Ils souhaitaient accéder directement à la réponse pour passer à l'activité suivante. Cela m'a permis de constater qu'ils étaient moins persévérants face aux difficultés rencontrées que le groupe de CM1-CM2. Lorsque des binômes de la classe de CM1-CM2 étaient bloqués sur une activité, ils n'étaient pas en demande d'une solution mais voulaient savoir quels outils ils devaient utiliser pour résoudre la situation ou ils cherchaient à trouver à quel endroit ils avaient fait une erreur qui empêchait la bonne exécution de leur programme.

5. Conclusion

La pratique d'activités de programmation en débranché a permis aux élèves d'aborder différemment les activités de déplacement d'un personnage sur un logiciel. Les élèves ont mobilisé les connaissances effectuées sur le langage du codage en manipulant avec aisance les commandes de base du logiciel et en cherchant à optimiser le programme par l'utilisation d'un élément plus complexe : la boucle de répétition.

Les situations d'enseignement apprentissage en débranché ont amené les élèves à faire évoluer leur posture d'un registre d'interprétation vers un registre de modification. Ils ont été capables d'adopter la posture de programmeur que je souhaitais leur faire atteindre à l'issue de cette séquence. Ils mettaient en place des stratégies pour utiliser de façon optimale le logiciel en ayant recours aux commandes à leur disposition et pouvaient raisonner sur leurs productions pour déceler des erreurs. Les élèves de la classe n'ayant pas pratiqué la programmation en débranché étaient dans un registre d'interprétation ou de participation. Ils avaient des difficultés à identifier eux-mêmes les erreurs de leur programme et avaient recours à l'aide d'un tiers pour réussir à dépasser les difficultés rencontrées.

La pratique des activités en débranché a eu l'effet d'homogénéiser le niveau global de la classe. Les élèves qui avaient des difficultés les ont surmontées lors des activités en débranché et ont eu plus d'aisance lors de la séance de réinvestissement sur l'outil numérique. Comme on l'a constaté, l'écart de la dernière activité atteinte par les binômes était beaucoup plus faible que dans l'autre classe n'ayant pas pratiqué d'activité de programmation en débranché.

Les élèves ont réinvesti des stratégies de coopération vues lors des séances en débranché. Si lors des premières séances en débranché certains travaillaient de façon individuelle, ils ont fait évoluer leur posture. Lors de la séance de réinvestissement, tous les élèves de cette classe ont su coopérer pour résoudre les situations alors que les élèves de l'autre classe observée travaillaient de façon individuelle au sein de leur binôme.

Après l'étude et la comparaison des deux groupes classes sur une activité de déplacement d'un personnage sur un logiciel de programmation, je constate que l'apprentissage en débranché a induit des comportements différents face aux activités branchées sur un logiciel. L'apprentissage de la programmation en débranché permet aux élèves de construire le sens du langage du codage tout en manipulant et en expérimentant par eux-mêmes ce qui leur permet une entrée facilitée dans les activités de programmation de déplacement d'un personnage sur un logiciel.

6. Bibliographie

Baron, G., Drot-Delange, B. (2016). « L'informatique comme objet d'enseignement à l'école primaire française ? Mise en perspective historique. » *Revue française de pédagogie*, 195(2), 51-62. disponible sur : <https://www.cairn.info/revue-francaise-de-pedagogie-2016-2-page-51.htm>.

Daval, N., Gravini, O. (2017). *Codage et programmation*. disponible sur : < <https://www.edumooov.com/fiche-de-preparation-sequence/83703/espace-et-geometrie/cm1-cm2/codage-et-programmation> >

Declercq, C., Tort, F. (2018). *Organiser l'apprentissage de la programmation au cycle 3 avec des activités guidées et/ou créatives*. RJC EIAH 2018, Apr 2018, Besançon, France. hal-01765408 disponible sur : < hal-01765408 >.

Declercq, C., Zeyringer, M. (2018) *Analyse de l'activité des élèves dans l'environnement PixelArt pour l'apprentissage de la séquence et de la répétition au cycle 3*. ETIC3, Jun 2018, Paris, France. disponible sur : <hal-01826065>

Fabius, L. (1985). *L'informatique pour tous*. Présentation à la presse, le 25 janvier 1985, disponible sur : <https://www.epi.asso.fr/revue/37/b37p023.htm>. >.

Grugier, O. (2018). « Rencontres entre des élèves de maternelle et des robots programmables. Des savoirs technologiques et informatiques ». In. *Colloque Etic3. école et Tic*. Paris Descartes du 27 au 28 juin 2018.

Lucas, J., Lucas, J.-C., Meunier, L., Trossevin, M.-P., Meunier, R. (2016). *Le nouvel à portée de Maths CM1*. Hachette éducation.

Lucas, J., Lucas, J.-C., Meunier, L., Trossevin, M.-P., Meunier, R. (2016). *Le nouvel à portée de Maths CM2*. Hachette éducation.

Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche (2018). *Bulletin officiel n°30 du 26 juillet 2018*. [en ligne] disponible sur : < https://cache.media.education.gouv.fr/file/30/05/0/ensel169_annexe2V2_986050.pdf >.

Ministère de l'éducation nationale, de l'enseignement supérieur et de la recherche. (2018) *Le numérique au service de l'École de la confiance*. [en ligne] disponible sur : < <https://www.education.gouv.fr/cid133192/le-numerique-service-ecole-confiance.html> >.

7. Annexes

Séquence détaillée

CM1 CM2	<i>Le déplacer sur un quadrillage</i>	<i>Espace et géométrie programmation</i>
<i>Compétences travaillées</i>	<p>Chercher (<i>domaine du socle : 2, 4</i>)</p> <ul style="list-style-type: none"> • s'engager dans une démarche, observer, questionner, manipuler, expérimenter, émettre des hypothèses, en mobilisant des outils ou des procédures mathématiques déjà rencontrées, en élaborant un raisonnement adapté à une situation nouvelle. • tester, essayer plusieurs pistes de résolution. <p>Modéliser (<i>domaine du socle : 1, 2,4</i>)</p> <ul style="list-style-type: none"> • utiliser les mathématiques pour résoudre quelques problèmes issus de situations de la vie quotidienne. <p>Représenter(<i>domaine du socle : 1, 5</i>)</p> <ul style="list-style-type: none"> • utiliser et produire des représentations de situations spatiales <p>Raisonner (<i>domaine du socle : 1, 2, 5</i>)</p> <ul style="list-style-type: none"> • résoudre des problèmes nécessitant l'organisation de données multiples ou la construction d'une démarche qui combine des étapes de raisonnement. • progresser collectivement dans une investigation en sachant prendre en compte le point de vue d'autrui <p>Communiquer (<i>domaine du socle : 1,3</i>)</p> <ul style="list-style-type: none"> • utiliser progressivement un vocabulaire adéquat et des notations adaptées pour décrire une situation, exposer une argumentation. • expliquer sa démarche ou son raisonnement, comprendre les explications d'un autre et argumenter dans l'échange 	
<i>Attendus de fin de cycle 2</i>	<p>Se repérer et se déplacer en utilisant des repères et des représentations</p> <ul style="list-style-type: none"> • se repérer dans son environnement proche ; • situer des objets ou des personnes les uns par rapport aux autres ou par rapport d'autres repères : <ul style="list-style-type: none"> ▀ vocabulaire permettant de définir des positions (gauche, droite, au-dessus, en-dessous, sur, nous, devant, derrière, près, loin, premier plan, second plan, nord, sud, est, ouest, etc.) ▀ vocabulaire permettant de définir des déplacements (avancer, reculer, tourner à droite/ tourner à gauche, monter, descendre, etc.) • s'orienter et se déplacer en utilisant des repères ; • réaliser des déplacements dans l'espace et les cadre pour qu'un autre élève puisse les reproduire ; • produire des représentations d'un espace restreint et s'en servir pour communiquer des positions ; • programmer les déplacements d'un robot ou ceux d'un personnage sur un écran : <ul style="list-style-type: none"> ▀ repères spatiaux ▀ relation entre l'espace dans lequel on se déplace et ses représentations 	
<i>Attendus de fin de cycle 3</i>	<p>Se repérer et se déplacer dans l'espace en utilisant ou en élaborant des représentations</p> <ul style="list-style-type: none"> • se repérer, décrire ou exécuter des déplacements, sur un plan ou sur une carte. • accomplir, décrire, coder des déplacements dans des espaces familiers. • programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation <ul style="list-style-type: none"> ▀ vocabulaire permettant de définir des positions et des déplacements (tourner à gauche, à droite, faire demi-tour, effectuer un quart e tour à droite, à gauche). 	
<i>Objectifs de la séquence</i>	<p>Se repérer sur un quadrillage Accomplir, décrire et coder des déplacements sur un quadrillage</p>	

	Titre de la séance	Objectif pour l'élève	Composante mobilisée	Durée	Matériel(s)
Séance 1	La bataille navale	Se repérer sur un quadrillage	Communiquer Représenter Modéliser	50'	jeu de bataille navale
Séance 2	Le robot idiot De l'espace réel à l'espace relatif	Accomplir et décrire des déplacements sur un quadrillage par téléguidage puis par de la programmation. Passer de l'espace réel à l'espace relatif	Raisonner Communiquer Représenter Chercher Modéliser	60'	plots
Séance 3	Espace relatif	Se repérer, décrire et exécuter les déplacements d'un personnage sur un quadrillage	Représenter Chercher	45'	
Séance 4	Faire correspondre un programme à une situation	Anticiper les déplacements d'un personnage sur un quadrillage	Raisonner Communiquer	30'	scènes + feuilles de programme
Séance 5	Faire correspondre un programme à une situation particulière sur un quadrillage	Anticiper les déplacements d'un personnage sur un quadrillage Utiliser les répétitions et les boucles pour coder un déplacement	Raisonner Communiquer	55'	personnages + scènes + feuilles de programme
Séance 6	Pixel Art	Programmer le déplacement d'un curseur sur un écran pour dessiner un motif	Chercher Raisonner Communiquer	55'	ordinateur

Remarque(s) : La séance 6 est une séance de réinvestissement sur un logiciel de programmation disponible sur ordinateur.

Résumé

Mots clefs : Enseignement de la programmation - programmation en débranché - cycle 3 - numérique - informatique

Résumé en français :

Dans le cadre de l'enseignement de l'espace et de la géométrie, il est attendu dans les programmes en vigueur à la rentrée 2018 que les élèves, en fin de cycle 3, soient capables de programmer le déplacement d'un robot ou d'un personnage sur un écran par l'utilisation d'un logiciel de programmation. L'entrée dans l'apprentissage de la programmation peut se faire par des situations en débranché sans recourir à des outils numériques. L'objectif de cette pratique est de proposer des situations de manipulation centrées sur les compétences à développer chez l'élève tout en favorisant son engagement dans la tâche. L'enseignement en débranché peut être une première entrée dans l'initiation à l'informatique. Nous faisons l'hypothèse que l'apprentissage de la programmation par des situations débranchées permettra aux élèves d'acquérir des connaissances et de construire des compétences spécifiques à la programmation qu'ils pourront réinvestir sur un outil numérique.

Keywords : The teaching of programming - unplugged - cycle 3 - digital - computer programming

Résumé en anglais :

In the context of the teaching of space and geometry, in the current programs (those in effect in 2018) pupils, by the end of cycle 3, should be able to program the moves of a robot, or of a character on screen with the use of a programming software. The introduction to the learning of programming can be done with unplugged situations without having to resort to digital tools. The aim of this, is to offer handling situations focusing on the skills the pupils should develop while enhancing their involvement in the task at stake. The unplugged teaching can be an introductory entry to computer programming. We assume that the study of programming through unplugged situations will enable the pupils to learn about and build the skills specific to programming that they will be able to invest back on digital tools.