

La place de l'erreur en programmation à l'école primaire

Julie Pocher

► To cite this version:

Julie Pocher. La place de l'erreur en programmation à l'école primaire. Education. 2019. dumas-02155261

HAL Id: dumas-02155261

<https://dumas.ccsd.cnrs.fr/dumas-02155261>

Submitted on 5 Nov 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.

La place de l'erreur en programmation à l'école primaire

Julie Pocher

**Sous la direction de
Christophe DECLERCQ**

**Master Métiers de l'Enseignement de l'Éducation et de la Formation
Mention Enseignement Premier Degré**

Mémoire de recherche :

**La place de l'erreur en programmation à
l'école élémentaire**

Julie Pocher

Remerciements

Je tiens à remercier en premier lieu mon directeur d'étude, Mr Christophe Declercq, pour son implication dans la réalisation de mon mémoire et le soutien apporté lors de cet écrit. Je le remercie également pour le prêt d'un robot Bluebot me permettant de réaliser ma séquence en classe et ainsi me permettre de récolter des données nécessaires à la réalisation de mon mémoire.

Je remercie également mes vingt-deux élèves de CE1 pour leur participation, leur implication et leur enthousiasme dans la réalisation de ce projet de programmation, ainsi que l'équipe enseignante de l'école Théodore Monod de Saint-Sébastien-sur-Loire pour leur soutien dans l'écrit de ce mémoire.

Je remercie particulièrement mes proches pour leur soutien nécessaire tout au long de l'écriture de ce mémoire et notamment mon amie Claire, qui a su me rassurer lorsque j'en avais besoin.

Enfin, je remercie l'École Supérieure du Professorat et de l'Éducation de l'Académie de Nantes ainsi que tout le corps enseignant qui nous accompagne au cours de la formation et plus particulièrement au cours de notre année de professeur stagiaire, riche en apprentissages et en questionnements.

SOMMAIRE

1.	Introduction	4
2.	Cadrage de la recherche	6
2.1	Recherches scientifiques	6
2.1.1	Pensée informatique et algorithme	6
2.1.2	Langage de programmation	9
2.1.3	De nombreuses compétences en jeu	9
2.2	Place de la programmation dans les ressources institutionnelles	12
2.2.1	L'évolution des programmes	12
2.2.2	Socle commun de connaissances, de compétences et de culture	14
2.3	La place de l'erreur dans les apprentissages	14
2.4	Diverses modalités d'enseignement de la programmation	17
2.5	Questions professionnelles de recherche	20
3.	Dispositif empirique de recueil de données	22
3.1	Ma séquence de programmation	22
3.2	L'observation de la place de l'erreur	23
4.	Analyse de données	25
4.1	Le robot Bluebot	25
4.2	L'application Tuxbot	32
4.3	La productivité du retour sur l'erreur	35
5.	Conclusion	39
	Bibliographie	41
	Annexe 1	43

1. Introduction

Dans le cadre de mon mémoire de deuxième année de master de l'enseignement, de l'éducation et de la formation, j'ai décidé d'aborder le thème de la programmation et du codage au cycle 2. L'informatique est un outil et un objet d'apprentissage qui fait sa première apparition dans les programmes de l'Education Nationale en 1985 en réponse au développement de l'informatique. La loi d'orientation et de programmation pour la refondation de l'école de la République, du 8 juillet 2013, réaffirme l'importance de l'utilisation du numérique à l'école primaire et en 2016, les nouveaux programmes font référence au langage informatique. La programmation et le code informatique font maintenant partie intégrante des attendus de fin de cycle, de la maternelle au lycée. Le domaine 1 du Socle commun intègre le langage informatique au même titre que les langages scientifiques et mathématiques. Cette importance donnée à l'informatique dans le cursus scolaire et le développement de compétences en programmation, transférables aux autres apprentissages, me donne envie d'approfondir mes connaissances et c'est dans ce cadre que j'ai choisi d'écrire un mémoire sur ce thème.

Je réalise cette année mon stage de professionnalisation à l'école Théodore Monod, à Saint-Sébastien-sur-Loire (44) dans une classe de CE1. Il s'agit d'une classe de 22 élèves ayant tous des profils différents avec une hétérogénéité des connaissances et des comportements importante. La mise en place de séquences nécessite donc une situation motivante qui leur permet d'entrer dans les apprentissages. L'utilisation du numérique comme objet d'apprentissage permet de développer de nombreuses connaissances transférables à d'autres disciplines. De nombreuses recherches montrent l'importance du numérique dans la résolution de problème et le développement de la pensée et c'est ce qui me permet de me poser de nombreuses questions. Steve Jobs affirmait que « chacun devrait savoir comment programmer un ordinateur parce que ça apprend à penser ».

Dans de ce mémoire il s'agira d'interroger la place et le rôle de l'erreur dans la programmation d'un déplacement, pour l'enseignant et pour les élèves et sa

productivité dans les apprentissages. Dans une première partie théorique, j'invoquerai les ressources scientifiques et institutionnelles et placerai la programmation dans le contexte scolaire en me focalisant sur l'erreur dans les apprentissages. Dans une deuxième partie, j'analyserai les données récoltées sur le terrain pour pouvoir répondre à ma question de recherche.

2. Cadrage de la recherche

2.1 Recherches scientifiques

L'informatique prend une part importante dans l'évolution de notre société et donc dans l'évolution des apprentissages en classe : on parle des technologies de l'information et de la communication (TIC). Il s'agit de sensibiliser les élèves à l'utilisation de ces nouvelles technologies ainsi qu'à l'usage qu'ils peuvent en faire. Le monde évoluant, l'école a pour but de préparer les élèves aux métiers de demain en favorisant la connaissance des outils numériques et leurs limites.

J'ai choisi dans ce mémoire de me concentrer sur l'apprentissage d'un langage propre à l'outil numérique : la programmation. Il convient donc de définir les termes afférents à l'informatique avant d'aborder la place que prennent les TIC dans les ressources institutionnelles. J'aborderai ensuite dans cette partie ce qui est au cœur de mon sujet : la programmation et la place de l'erreur dans son apprentissage.

2.1.1 Pensée informatique et algorithme

Dans son article « Computational Thinking » publié en 2006 dans sa version originale et traduit en Français par Pierre Lescanne en 2008, Jeannette M. Wing marque l'importance de la **pensée informatique** dans la résolution de problèmes, transférable à d'autres disciplines. Pour ce professeur d'informatique à l'université de Carnegie-Mellon (USA), la pratique de la pensée informatique devrait être intégrée aux apprentissages de tous les écoliers et ce dès la maternelle. Elle énonce qu' « *adopter un mode de pensée informatique conduit à résoudre des problèmes, à concevoir des systèmes et à comprendre le comportement humain différemment, en s'appuyant sur les concepts fondamentaux de la discipline informatique* ». La connaissance de la pensée informatique est pour elle une compétence nécessaire dans la compréhension du monde en particulier dans la compréhension de la différence entre l'intelligence humaine et l'intelligence des machines : l'homme réfléchit et conçoit, la machine exécute. C'est dans ce sens que la part de l'erreur est importante puisque l'ordinateur ne peut pas déceler un bogue, seul l'humain peut le repérer après exécution par la

machine et ainsi le réparer. J. Wing différencie donc l'intelligence mécanique et l'intelligence humaine par la réflexion et l'exécution qui s'en suit.

Didier Roy, chercheur membre de l'équipe Flowers d'Inria Bordeaux Sud-Ouest, traite la question de la pensée informatique à l'école et son importance dans la compréhension de comportements humains. Dans son chapitre « la robotique éducative au service du développement de la pensée informatique – exemple de deux dispositifs open source », D. Roy met en évidence l'importance de la robotique dans le développement de la pensée informatique en prenant en exemple deux robots. Celui-ci a travaillé avec de nombreux chercheurs et enseignants pour mettre en évidence le rôle du robot dans le développement de nombreuses compétences telles que l'apprentissage de la démarche scientifique, la capacité à débattre et exprimer son point de vue, la compréhension... L'usage de la robotique permettrait selon lui de construire des attitudes et des connaissances universelles nécessaires à la résolution de problème. Il évoque d'une part le rôle de l'erreur dans le travail de recherche qui met l'élève en position de chercheur. En effet l'élève observe directement l'effet de sa programmation sur le robot et peut ainsi réajuster son travail sans peur de l'échec. Les apprentissages sont alors impactés positivement puisque les élèves sont acteurs de leur réussite, ce qui induit une plus grande motivation. D'autre part, il relate la coopération des élèves dans le travail de recherche qui développe des compétences d'argumentation et l'emploi des débats.

Par ailleurs, la recherche de programmation à l'aide de robot fait intervenir des compétences scientifiques en mettant en avant la démarche d'investigation. Selon Didier Roy, « la robotique paraît être un formidable champ de découverte et d'application de la pensée informatique¹ ».

La **pensée algorithmique** est définie par Alexander Repenning (2014) comme une étape de la pensée informatique qui permet de diviser en tâches simples un programme que l'on souhaite faire exécuter par une machine. Il s'agit de décrire le processus utilisé dans la réalisation d'un programme pour permettre à une machine ou un humain de le réaliser. Cette étape à la pensée informatique est donc nécessaire dans la compréhension du processus de création du programme

¹ D. Roy, 2017

et également dans sa réalisation par une machine ou un humain. Pierre Tchounikine précise qu'il ne faut pas réduire la pensée informatique à l'algorithmique mais l'aborder comme la résolution d'un problème conduisant à l'écriture d'un algorithme amenant parfois à un programme exécutable par une machine : « *la pensée informatique ne se réduit donc pas à l'algorithmique mais le concept d'algorithme est au cœur de la pensée informatique*² ». Le schéma ci-après résume la pensée informatique et l'implication de la pensée algorithmique (Peggy Delarbre, 2017).

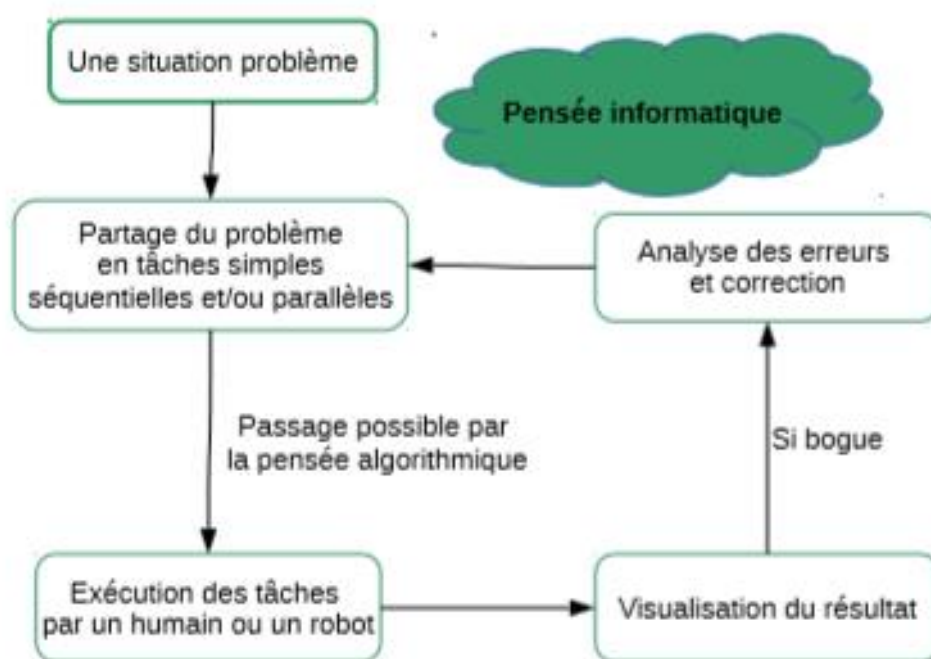


Figure 1 - La pensée informatique

D'après Gilles Dowek, informaticien et logicien français, « *un algorithme est une recette qui permet de résoudre un certain problème de manière systématique*.³ » Il explique que cette notion n'est pas propre à l'informatique puisqu'elle est développée depuis longtemps dans les mathématiques pour résoudre des problèmes.

La création d'un algorithme ne suffit pas à une machine pour réaliser ce qui lui est demandé, seule l'écriture de cet algorithme, appelé langage de programmation permet de réaliser cette tâche.

² Pierre Tchounikine, 2016

³ Dowek, 2011

2.1.2 Langage de programmation

La programmation est l'utilisation d'un langage permettant de coder un algorithme compréhensible par un ordinateur ou un humain dans le but de le réaliser. Ce langage doit donc être connu par tous pour pouvoir être décrypté par une machine programmable. Dans la programmation de robot, le langage utilisé correspond à un déplacement représenté par des flèches. Ces flèches permettent de donner les indications suivantes : avancer, tourner à droite, tourner à gauche, reculer. Un programme ainsi écrit sur la machine permet au robot de se déplacer en ayant connaissance de ce qui lui est demandé.

Les notions d'algorithmes et de programmation sont comparées par Gilles Dowek à une recette, comme énoncé plus haut : « *Une recette peut être exécutée et même être transmise de génération en génération, sans être écrite, ni même verbalisée. C'est cette seconde notion qui correspond à la notion d'algorithme. Dès que la recette est écrite ou verbalisée, elle doit être comparée, non à la notion d'algorithme, mais à celle de programme*⁴.

Le langage de programmation est donc un langage qui permet de rendre reproductible et stable un algorithme.

2.1.3 De nombreuses compétences en jeu

L'apprentissage de la programmation permet de développer de nombreuses compétences nécessaires à l'acquisition des compétences du socle commun en fin de scolarité obligatoire. Il est important de raisonner en termes d'objectifs pédagogiques pour savoir ce que l'on attend des élèves et ainsi développer des compétences numériques sans utiliser le numérique pour sa seule fonction d'outil. Au-delà des compétences de programmation et de repérage dans l'espace énoncées dans les programmes, elle développe des capacités langagières dans la mise en commun ou l'explication des procédures. En effet, par la découverte d'outils peu utilisés en classe, les élèves résolvent des problèmes qui leur sont posés. Ils entrent donc dans une démarche de recherche. Par ailleurs, la

⁴ Dowek, 2011

pédagogie utilisée par l'enseignant permet de développer des compétences de coopération importantes dans la socialisation des élèves. Le partage des connaissances est un outil important qui permet de confronter les points de vue garantissant la formation de futurs citoyens.

La programmation permet également de développer la notion d'algorithme qui permet selon Piaget de développer la logique des enfants et ainsi leur esprit d'analyse. Le langage de programmation aussi appelé « codage » n'est pas inné et son apprentissage et sa conceptualisation nécessitent une phase de manipulation de l'objet avant d'en comprendre le sens. G. L. Baron et B. Drot-Delange manifestent l'importance de la situation didactique dont s'empare l'enseignant pour aborder une séquence sans passer à côté des notions numériques en jeu⁵. Bien que le rôle de l'enseignant soit plus un rôle d'entraîneur, il est important que la situation à laquelle les élèves sont confrontés soit pensée en termes d'objectifs didactiques. Grugier développe également cette idée en analysant une séquence de programmation en classe de maternelle. Il met en avant les compétences acquises au cours de cette séance mais ne néglige pas le rôle de l'enseignant qui est primordial : *« ces robots programmables peuvent être pris en charge par des élèves de l'école maternelle. Ces robots permettent de faire acquérir la compréhension du fonctionnement d'un objet et du vocabulaire nouveau (...). Cependant, le rôle de l'enseignant pendant la découverte va conditionner les apprentissages. »*⁶

O. Grugier réalise une grille d'observation permettant d'analyser les comportements des élèves face à l'objet programmable. Cette grille repose sur quatre registres d'actions sur un objet programmable en fonction du degré d'« utilisation » et d'« action instrumentée ». Il reprend le travail de Martinand (1994) *« qui suggère une évolution du concept en introduisant quatre registres de technicités. »* Grugier précise que *« chaque registre comporte des compétences, des représentations et des valeurs spécifiques. Ces registres sont pensés d'un point de vue d'une construction curriculaire notamment pour orienter l'activité des enseignants. »*

⁵ G. L. Baron et B. Drot-Delange, 2016.

⁶ Grugier, 2018.

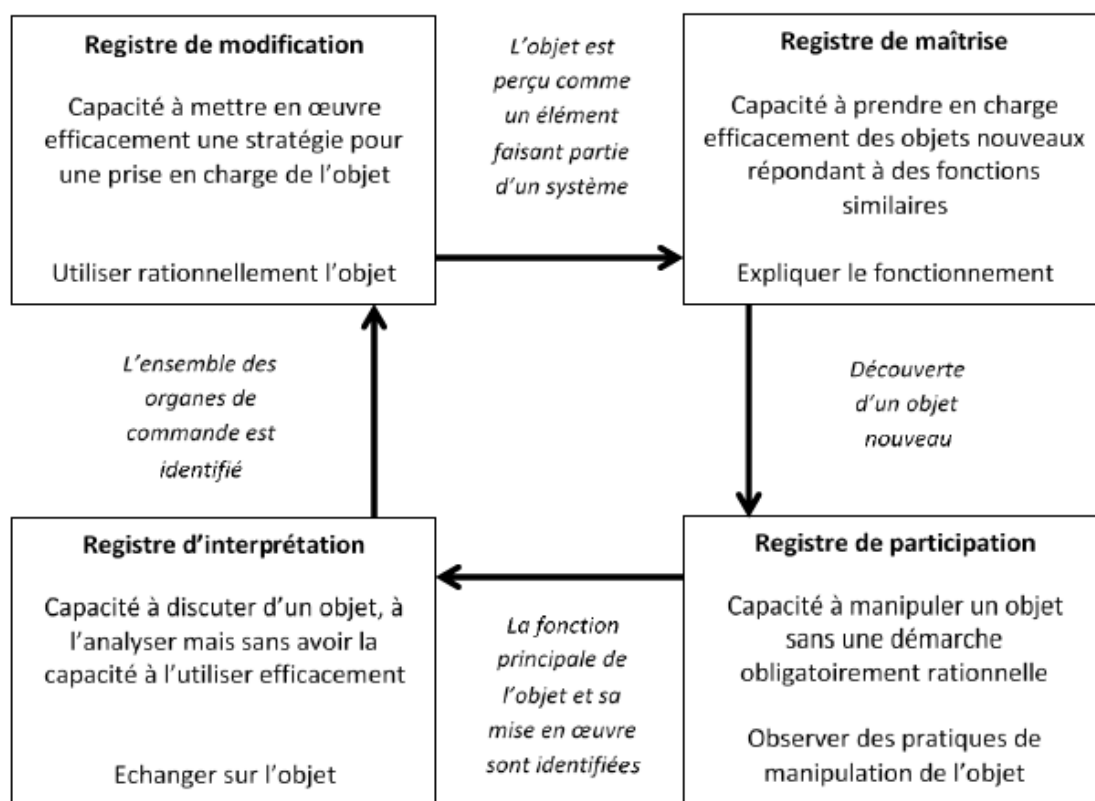


Figure 2 - Grille d'analyse de la technicité mise en jeu dans les moments d'enseignement-apprentissage proposés, Grugier (2017)

Cette grille d'analyse montre bien l'évolution du comportement des élèves dans la réalisation d'un problème posé mettant en jeu la programmation.

Michel Spach, enseignant chercheur, à l'université Paris Descartes, rejoint le point de vue de Grugier en mettant en avant le développement des compétences chez des élèves de primaire et conforte le rôle déterminant de l'enseignant. Il ajoute à ses recherches l'importance d'une situation problème qui permet aux élèves d'entrer dans une démarche de recherche. Il accorde également une part de ses recherches à l'importance de l'erreur dans la recherche du fonctionnement des robots.

« Les élèves sont incités à chercher, tester, justifier, optimiser, mener une réflexion sur les erreurs rencontrées parfois mises en évidence par la présence d'obstacle didactique. La recherche de correction à apporter à ces erreurs favorise le développement de démarches expérimentales par l'anticipation d'un résultat, l'émission et la validation d'hypothèses, le tâtonnement, etc.⁷ »

⁷ Spach, 2018.

La programmation élargit le champ des compétences en plaçant l'erreur au cœur des apprentissages en permettant aux élèves de se tromper pour comprendre. L'utilisation de robot programmable est dans ce sens un outil important qui permet de se confronter à l'erreur.

2.2 Place de la programmation dans les ressources institutionnelles

Les ressources institutionnelles accordent une place de plus en plus importante à la programmation notamment depuis la loi de refondation de l'école. La loi d'orientation et de programmation pour la refondation de l'Ecole de la République, du 8 juillet 2013, entraîne de nombreux changements dans le domaine du numérique par la création du Socle commun, la publication de nouveaux programmes et l'instauration d'un service public du numérique éducatif. Par ailleurs, la mise en place des écoles supérieures du professorat et de l'éducation (ESPE) revoit la formation des enseignants en intégrant, entre autres, les enjeux et les usages pédagogiques du numérique dans la formation initiale et continue. Il s'agit de mieux former les enseignants au numérique pour permettre la mise en place d'apprentissages en classe respectant aux mieux les attendus des nouveaux programmes. L'intérêt est de faire évoluer les pratiques des enseignants dans le sens des évolutions informatiques actuelles.

2.2.1 L'évolution des programmes

Le numérique à l'école fait son apparition dans les programmes de 1985 mais se développe plus particulièrement dans les programmes de 2016 avec des attendus spécifiques à chaque cycle. La programmation en est un exemple. En cycle 1, les programmes stipulent que les élèves doivent être capables d'« identifier le principe d'un algorithme et poursuivre son application » dans le champ « *construire les premiers outils pour structurer sa pensée*⁸ » et d'« *utiliser des*

⁸ Programme de maternelle, MEN, 2016

objets numériques » dans le domaine « explorer le monde ». Il s'agit d'une première étape dans la construction de la pensée informatique.

Pour poursuivre en cycle 2, le programme de mathématiques précise que les élèves doivent être capables de « *coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran*⁹ ». Dans ce sens, la programmation d'un robot ou d'un personnage sur un écran est citée comme exemple de situation d'apprentissage.

Au cycle 3, la programmation intervient dans deux champs d'apprentissage. Elle intervient en premier lieu dans les sciences et la technologie pour « *concevoir et produire tout ou partie d'un objet technique en équipe pour traduire une solution technologique répondant à un besoin*¹⁰ », en modélisant « *virtuellement à travers des applications programmables permettant de visualiser un comportement* ». Dans ce même champ d'apprentissage, les « *notions d'algorithmes [et d']objets programmables* » font l'objet d'une étude. Dans un deuxième temps, la programmation intervient en mathématiques avec comme attendu de « *programmer les déplacements d'un robot ou ceux d'un personnage sur écran* » en utilisant le vocabulaire adapté.

Les programmes montrent une évolution des apprentissages du numérique comme objet et non outil. Il s'agit bien d'apprendre à utiliser le numérique dans une ère où l'informatique prend toute sa place. L'éducation nationale met en avant cette révolution sur le site « devenir enseignant » :

« La révolution numérique est une chance pour l'École car les nouveaux outils offrent un potentiel de renouveau pédagogique important, pouvant améliorer l'efficacité et l'équité du système éducatif. Elle est aussi un défi lié au développement rapide des usages du numérique : dans un monde où l'outil informatique et les réseaux sociaux occupent une place centrale, il est essentiel que l'école donne aux élèves les savoirs correspondants, qu'elle les prépare à la citoyenneté numérique et à l'emploi de demain¹¹ ».

⁹ Programmes, MEN, 2015

¹⁰ Ibid

¹¹ MENJ, « L'école change avec le numérique », site officiel de l'éducation nationale. <http://www.devenirenseignant.gouv.fr/pid34343/l-ecole-change-avec-numerique.html>

2.2.2 Socle commun de connaissances, de compétences et de culture

Le socle commun de connaissances, de compétences et de culture intègre le numérique comme langage pour penser et communiquer dans le domaine 1. A la fin de sa scolarité, l'élève « connaît les principes de base de l'algorithme et de la conception des programmes informatiques [et] les met en œuvre pour créer des applications simples¹² ». Le domaine 2 met en avant l'utilisation d'objets numériques dans la réalisation de projets au service de la collaboration et du travail de groupe. Le socle commun montre toute l'importance accordée au numérique dans la construction d'un citoyen autonome et responsable.

2.3 La place de l'erreur dans les apprentissages

Gérard de Vecchi, enseignant-chercheur et formateur expliquent dans son livre « Aider les élèves à apprendre » la place importante de l'erreur dans les apprentissages.

« On ne peut apprendre que lorsqu'on ne connaît pas. Quand un élève ne se trompe pas... c'est qu'il sait. S'il sait, il n'apprend pas réellement... donc il perd son temps ! Et c'est bien quand il commet une erreur qu'il va pouvoir changer... donc apprendre ! L'erreur doit donc avoir un statut positif dans les apprentissages¹³. »

L'erreur fait partie intégrante du processus d'apprentissage et constitue un levier d'appui dans la construction des savoirs. L'élève procède par tâtonnements, essais, erreurs pour accéder aux notions. Le retour réflexif sur l'erreur est nécessaire aux élèves et devient un outil collectif de stabilisation du savoir. Les programmes de 2015 font référence à la place de l'erreur en modifiant l'idée de « faute » qui dominait auparavant. L'erreur est bien ici considérée comme une aide aux apprentissages et au développement des compétences et des savoirs. Il s'agit de développer l'autonomie des élèves et de créer une relation de confiance au sein de l'école en mettant l'élève en position haute : il doit être créateur et récepteur du savoir.

¹² MEN, 2015.

¹³ Gérard de Vecchi, 2014.

L'erreur est un outil pour l'enseignant qui peut ainsi réguler les apprentissages et modifier ses modalités de travail pour permettre à chaque élève de réussir tout en comprenant son erreur et en l'analysant. C'est dans cette continuité que se trouve tout l'enjeu du professeur des écoles dont l'une des compétences du référentiel commun à tous les professeurs est d'« évaluer les progrès et les acquisitions des élèves [en] analys[ant] les réussites et les erreurs ». Il doit donc être en mesure d'identifier les démarches d'apprentissages des élèves pour comprendre les processus d'apprentissage. L'enseignant peut alors, par son analyse, différencier les approches pédagogiques de plusieurs manières : créer des groupes de niveaux, simplifier ou complexifier la tâche, mettre à disposition du matériel, mettre en place le tutorat entre élèves... Il s'agit bien de rendre l'erreur productive et de lui redonner toute sa place dans les apprentissages.

L'erreur est également un outil pour l'élève. La place de l'erreur prend toute son sens dans le socle commun de connaissances, de compétences et de culture dans le but de favoriser la confiance des élèves. Le domaine 2 « méthodes et outils pour apprendre » développe cette idée en indiquant les compétences suivantes : « *savoir identifier un problème, s'engager dans une démarche de résolution, mobiliser les connaissances nécessaires, analyser et exploiter les erreurs, mettre à l'essai plusieurs solutions, accorder une importance particulière aux corrections*¹⁴ ». Dans le domaine 4 « systèmes naturels et systèmes techniques » on retrouve des compétences en lien avec la démarche d'investigation : « *formuler des hypothèses, les tester et le éprouver* », « *manipuler, explorer plusieurs pistes, procéder par essais et erreurs*¹⁵ ».

C'est dans cette logique que la démarche d'investigation, développée en sciences et étendue aux autres enseignements, prend tout son sens. L'élève doit chercher, modéliser pour apprendre. Dans l'apprentissage du numérique, l'erreur tient toute son importance puisque les élèves apprennent grâce à l'expérimentation qui passe le plus souvent par des procédures d'essais-erreurs qui permettent de résoudre un problème posé. On peut voir dans cette méthode le principe de la démarche d'investigation qui place l'élève au cœur de ses

¹⁴ Socle commun de connaissance, de compétences et de culture, 2015

¹⁵ Ibid

apprentissages. Cette démarche place l'élève en position haute en lui proposant un problème qu'il devra résoudre. Il est en mesure d'émettre des hypothèses et de construire ses propres connaissances tout en raisonnant. La démarche peut se décliner suivant différentes étapes :

- Une situation déclenchante qui doit favoriser la curiosité des élèves, susciter son intérêt et amener l'élève à se poser des questions
- La formulation d'hypothèses pour répondre au questionnement qu'entraîne la situation déclenchante ; il s'agit de recueillir les représentations des élèves, souvent en dictée à l'adulte en cycle 1.
- La réalisation d'expériences pour tester les différentes hypothèses. Cette étape permet aux élèves de s'approprier les connaissances en testant, expérimentant, vérifiant les questions qu'ils se posaient. La manipulation est primordiale en cycle 1.
- L'observation des résultats pour en tirer des conclusions.
- Construction et structuration des connaissances en réalisant une synthèse orale ou écrite pour garder une trace des recherches effectuées.

Dans l'apprentissage du numérique une démarche d'investigation n'est pas suffisante et il faut penser à introduire une pédagogie de projet qui permettra de développer de nombreuses compétences transversales telles que la coopération, le travail en équipe, la répartition du travail, les débats et l'argumentation... Cette démarche de travail augmente la réussite des élèves en difficultés par l'entraide et le tutorat des élèves ayant plus de facilités. Le Conseil National du Numérique publie en 2014, dans un rapport sur le numérique, que l'informatique pourrait être une voie de retour motivante sur le chemin de la connaissance pour les élèves « décrocheur ».

« Les élèves y découvrent le plaisir d'essayer, de se corriger, d'apprendre avec les autres, le plaisir de transmettre un savoir dont ils sont fiers. Cela provient à la fois de l'intérêt de la matière, et d'un enseignement en « mode projet » qui donne un but, qui valorise l'essai-erreur, qui invite à la collaboration, qui redonne confiance en soi et assoit une compréhension plus solide¹⁶ »

¹⁶ CNN, 2014.

L'erreur dictée par une machine permet de dédramatiser l'échec pour au contraire en faire une force de travail. Lorsque celle-ci est constatée par l'enseignant, un élève en difficulté peut se sentir en situation d'échec et ne pas vouloir revenir sur son travail. Dans un article publié dans L'Obs, la journaliste Emilie Brouze précise que les élèves sembleraient plus confiants à essayer un exercice sur un outil numérique sans peur de se tromper. Interviewé dans Rue89, André Tricot affirme que « *pour un élève en difficulté scolaire, la répétition de retours négatifs sur son travail peut être perçue comme une menace. Le fait qu'ils soient exprimés par une machine gomme cette forme de pression sociale*¹⁷ ».

Il est donc important de comprendre le rôle que tient l'erreur dans les apprentissages, du côté des élèves et de l'enseignant, ainsi que dans toutes les matières enseignées à l'école. Pour reprendre les mots de Jean-Pierre Astolfi, « *l'erreur est un outil pour enseigner* ».

2.4 Diverses modalités d'enseignement de la programmation

L'enseignement de la programmation permet de développer de nombreuses compétences dans le cadre des programmes. Il existe différentes manières de l'enseigner qui développeront des compétences et des objectifs divers, un des objectifs premiers en programmation étant de comprendre la notion d'algorithme et de pouvoir l'utiliser (séquence d'instructions, valeur variable, test et boucle). Peut-importe la méthode utilisée, il est important de travailler en termes de pédagogie de projet pour permettre aux élèves d'entrer dans une démarche de recherche : apprendre à raisonner, résoudre des problèmes, entrer dans une démarche d'investigation, développer la logique, structurer sa pensée.

Pour ce faire on peut utiliser les activités branchées ou les activités débranchées.

Les activités débranchées ou *Computer Science Unplugged* correspondent à un apprentissage sans objet numérique. Ces activités ne nécessitant pas d'outillage particulier, il est donc plus facile pour les professeurs des écoles de le mettre en

¹⁷ Emilie Brouze, L'Obs avec Rue 89, 27 avril 2016.

place dans les classes. Le plus souvent sur papier ou sous forme de jeux de plateau, elles permettent d'initier aux concepts informatiques de manière ludique et attractive. L'intérêt de cette méthodologie de travail est d'amener les élèves à penser par eux-mêmes et à coopérer. Béatrice Drot-Delange (2013) développe l'intérêt des activités débranchées dans l'apprentissage de l'outil informatique dans son article « *Enseigner l'informatique débranchée : analyse didactiques d'activités* ». Selon cette enseignante-chercheur de l'Université Clermont Auvergne, des activités débranchées passant par le jeu et la manipulation sont nécessaire à la motivation et l'enrôlement des élèves dans des tâches complexes et la construction des concepts informatiques. Il est important que les élèves soient en situation de recherche pour leur permette d'atteindre les objectifs d'apprentissage visés : « *ils sont amenés à découvrir les réponses par eux-mêmes, l'objectif n'étant pas d'enseigner des réponses, mais de faire « jouer » avec les concepts.*¹⁸ »

Lors des activités numériques, il est important de se situer dans la zone proximale de développement des élèves, énoncée par Vygotsky, pour leur permettre de réussir l'activité tout en proposant une certaine difficulté garantissant la recherche et la construction des concepts. Il s'agit bien de créer des situations motivantes qui ne soient pas décourageantes pour les élèves.

Le jeu du robot idiot est une des activités débranchées exploitée lors de l'apprentissage du langage de programmation. Il permet d'initier à la notion d'algorithme en faisant agir les élèves de manière corporelle pour suivre ou créer un programme. Ce jeu permet de passer d'un langage oral à un langage écrit en impliquant des variables didactiques pour comprendre le rôle d'un programme et l'enjeu du code utilisé. Il s'agit de donner du sens à la notion d'algorithme par une situation vécue et de comprendre le rôle du robot qui ne fait que suivre les instructions donnés par le programmeur. Le passage par le corps est souvent nécessaire à certains élèves pour mettre du sens sur les apprentissages. Toute activité vécue par l'élève permettra de développer des compétences plus stables et précises.

¹⁸ Beatrice Drot-Delange (2013).

Les activités branchées quant à elles correspondent à des activités qui impliquent l'utilisation de l'outil numérique. Ces activités qui mettent en avant la machine permettent aux élèves de comprendre l'enjeu d'un programme codé correctement qui sera reproduit par une machine. L'exemple du robot Bluebot est ici celui qui sera étudié. La motivation des élèves semble être plus effective lorsqu'ils obtiennent une réponse immédiate sur le travail fourni. En effet, lorsqu'un programme est réalisé sur un robot, on observe directement le déplacement du robot sur le quadrillage, il est ainsi plus facile de comprendre les bogues de la machine. La machine réalisant le programme donné par l'humain, ici l'élève, les erreurs réalisées sur le parcours pour atteindre le but donné (par exemple une case du quadrillage) sont immédiatement reconnaissables. Le programmeur peut alors identifier et modifier son erreur. L'erreur est prise en compte pour réussir comme le stipule les nouveaux programmes. L'erreur est au service de l'apprentissage.

Dans une recherche récente, Spach (2018) s'est intéressé au processus d'apprentissages engagés par les élèves lors d'une séquence de programmation de robots Bee-bots. Cette étude se centre sur les enjeux éducatifs qu'implique la compréhension de nouveaux concepts et leur rôle dans les apprentissages à l'école. Il s'interroge sur la notion de « pause » en programmation en mettant en avant le rôle de l'enseignant dans « l'accompagnement pédagogique [et] l'adaptation des artéfacts didactiques » en lien avec l'informatique.

« La mise en œuvre du scénario s'opère en prenant appui sur des gestes professionnels éprouvés (explicitation de consignes, soutien aux activités de recherche, institutionnalisation des savoirs, etc...), confortant le rôle déterminant de l'enseignante dans l'organisation des séances, le choix des modalités ainsi que dans la régulation des échanges¹⁹. »

L'enseignant crée des situations qui nécessitent l'action de la touche pause dans la programmation pour permettre aux élèves d'être en situation de recherche quant à son utilité et son fonctionnement. En effet, contrairement à son action habituellement directe sur un programme, la touche pause sur le robot intervient dans le programme. Il s'agit d'une pause écrite dans le

¹⁹ Spach, 2018.

programme et non pas la mise en pause de tout le programme. Les élèves développent différents schèmes qui les amènent à réfléchir sur l'écriture de leur programme et à les modifier. L'erreur est donc ici importante dans la compréhension du fonctionnement du robot et « *l'activité de débogage de programme dans le cas de déplacement synchronisé de deux robots est favorisée par la visibilité de l'erreur qui prend la forme d'une collision*²⁰. »

C'est en observant le rôle des activités branchées et leur importance dans le rôle de l'erreur comme outil d'apprentissage en numérique que j'ai décidé de réaliser une séquence portant sur la programmation d'un robot pour en analyser la productivité de l'erreur en programmation.

2.5 Questions professionnelles de recherche

La place de l'erreur est une donnée nécessaire dans le processus d'apprentissage des élèves et notamment dans l'apprentissage de la programmation comme je l'ai constaté de plusieurs auteurs cités plus haut. Pour répondre aux différentes hypothèses que je me suis posées, j'ai récolté des données dans ma classe de CE1 permettant d'analyser de nombreuses situations d'apprentissage.

Je me demande tout d'abord dans quelle mesure l'erreur est-elle productive chez les élèves ? L'erreur est-elle utilisée comme un outil essai-erreur ou permet-elle de modifier le programme construit ? Le retour de l'exécution du robot donne-t-il une information précise à tous les élèves sur l'endroit du programme où ils se sont trompés ? Il s'agit de comprendre si l'élève interprète son erreur pour modifier son programme en incrémentant le nouveau programme ou l'interprète seulement comme une erreur de son programme qu'il doit reprendre depuis le début en oubliant le premier réalisé.

J'observerai également la place que tient l'erreur avec l'objet tangible, le robot, et lors de l'utilisation du programme sur ordinateur avec l'application Tuxbot. S'il existe une différence, dans laquelle de ces deux situations les modalités de retour

²⁰ Ibid.

sur l'erreur sont-elles concrètes ? Dans laquelle de ces deux situations d'apprentissages l'erreur est-elle plus productive ?

L'erreur permet-elle aux élèves de progresser dans l'acquisition de l'objectif de programmation ?

Toutes ces questions permettent d'analyser la place de l'erreur dans la programmation et sa productivité. Pour cela j'ai mis en place une séquence de programmation mettant en jeu l'outil robot et l'application Tuxbot sur ordinateur.

3. Dispositif empirique de recueil de données

3.1 Ma séquence de programmation

Je réalise ma séquence²¹ de programmation dans ma classe de CE1 constituée de vingt-deux élèves. Cette séquence s'inscrit sur différents ateliers : la programmation sur ordinateur à l'aide de l'application Tuxbot, la programmation sur papier et la programmation sur le robot Blue-bot. Il s'agit d'atteindre l'objectif suivant : « *coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran*²² ». Plusieurs séances seront nécessaires à la réalisation de cet objectif final, le but étant de passer par plusieurs étapes pour permettre aux élèves de conceptualiser cette notion de programmation souvent difficile. J'ai décidé de débiter la séquence par une approche corporelle à l'aide du jeu du robot idiot, permettant aux élèves de vivre des situations concrètes dans lesquelles ils sont acteurs. Cette organisation de travail a permis aux élèves de se rendre compte de la difficulté de guider une personne sans utiliser le langage et ainsi découvrir le codage nécessaire pour programmer un déplacement.



²¹ Séquence en Annexe 1.

²² Programmes cycle 2, MEN, 2015.

Suite à cette approche, j'ai décidé de commencer par des activités débranchées avant d'entamer les activités branchées. Comme l'annonçait Grugier, il est important de passer par la manipulation pour permettre aux élèves de comprendre comment fonctionne le robot et en quoi consiste la programmation d'un algorithme. Les activités débranchées restaient difficiles pour certains élèves qui avaient du mal à programmer les déplacements notamment avec les flèches « tourner » qui ne réalisent pas de déplacement avant en même temps. Pour cette difficulté, seul le robot leur permet de comprendre cette notion en réalisant l'action de leur programmation. C'est après cette étape que j'ai mis en place les activités branchées. Tout d'abord par soucis d'organisation matérielle, n'ayant le robot qu'à la fin de cette séquence mais aussi pour permettre aux élèves de CE1 d'observer le rôle de leur erreur. Source de motivation, les robots ont été une étape clé de ma séquence et celle-ci que j'ai souhaité observer.

3.2 L'observation de la place de l'erreur

J'ai décidé d'observer le rôle et la place de l'erreur en programmation chez les élèves de ma classe. Certains de mes élèves ayant beaucoup de mal avec l'échec, j'ai trouvé intéressant de m'intéresser au rôle que celle-ci pouvait avoir lorsqu'elle est induite par un objet et non par l'enseignant. Pour ce faire, je me suis focalisée sur les activités branchées. Lors d'une première étape, j'ai observé la programmation du robot sur quadrillage sans anticipation. Les élèves devaient déplacer le robot d'un point de départ à un point d'arrivée, avec ou sans contrainte. Dans un deuxième temps, j'ai décidé d'observer la création d'un programme avec anticipation : l'algorithme était écrit par les élèves à partir d'un quadrillage papier avant d'être vérifié à l'aide du robot. C'est le lien entre l'écrit et le robot que j'ai décidé d'analyser pour questionner la productivité de l'erreur chez les élèves.

Les élèves travaillaient par groupe de deux ; ils disposaient d'un quadrillage papier sur lequel ils devaient programmer le déplacement du robot pour écrire des mots donnés. Une fois le programme écrit, les élèves venaient le tester sur le quadrillage au sol en programmant le robot. L'observation de son déplacement permettait de confirmer ou d'infirmer le programme créé. C'est prioritairement cette étape que j'ai décidé d'observer.

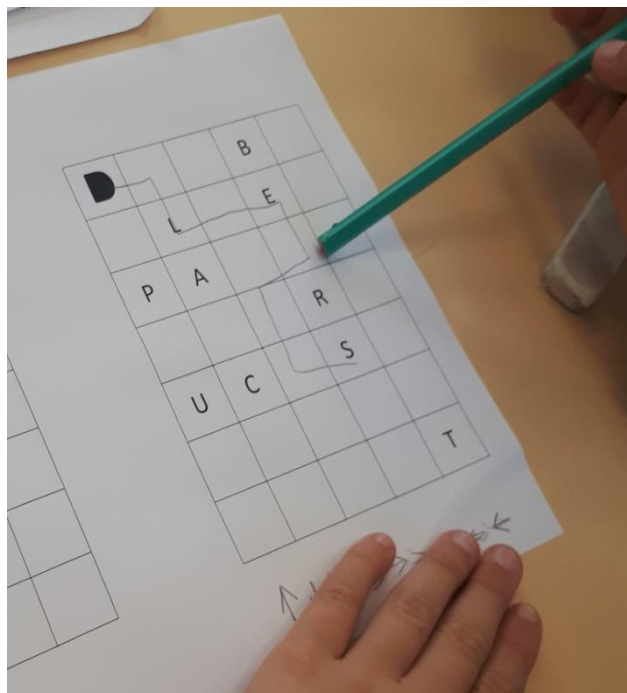


Figure 3 - codage de mots

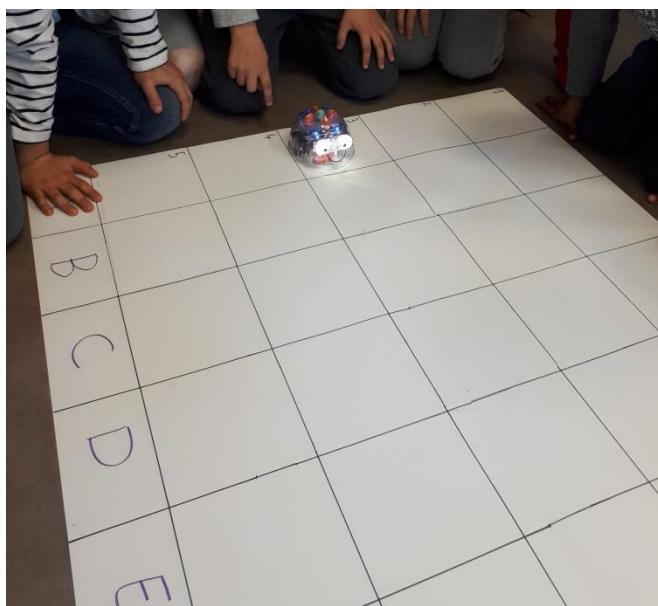


Figure 4 - programmation du robot sur quadrillage au sol

Pour analyser le rôle et la place que tient l'erreur dans l'apprentissage de la programmation, j'ai réalisé des vidéos montrant les réactions des élèves face aux erreurs produites par le robot suite à leur programmation. Les vidéos n'étant pas facilement transcribibles, j'utiliserai des réactions d'élèves ou des phrases dites lors des captations. Je transcrirai certaines vidéos qui me semblent nécessaires, notamment lorsqu'un élève m'explique toute sa démarche. En plus de ces vidéos, j'analyserai l'action des élèves grâce à des photos prises en classe.

4. Analyse de données

Les expérimentations qui ont été menées tenteront de répondre aux différentes hypothèses que je me suis posées.

4.1 Le robot Bluebot

Tout d'abord l'utilisation du robot dans un contexte de programmation est indéniablement source de motivation. Chaque élève a adhéré au projet avec enthousiasme et tous souhaitent manipuler le robot. La programmation sur papier motivait les élèves ayant plus de facilité tandis qu'elle semblait être un obstacle pour les élèves en difficulté qui se démotivaient devant la tâche à accomplir. En effet, étant motivés par le robot qui ressemblait à « un jeu » pour certains élèves, ceux-ci ne voulaient pas passer par l'écrit pour comprendre ce qu'ils faisaient. Il semblerait que la phase écrite est une étape de décentration très difficile pour certains élèves qui ont du mal à se positionner à la place du robot mais nécessaire à la réalisation des objectifs. Plusieurs difficultés à cela que l'on retrouve également lors de l'utilisation du robot :

- L'action de la plateforme

Les flèches sur la plateforme représentent des actions données plus haut. En effet, quatre flèches représentent les actions suivantes : « AVANCER », « RECULER », « TOURNER A DROITE », « TOURNER A GAUCHE », permettant au robot de se déplacer. Un autre bouton « PAUSE » sert à programmer un arrêt lors du déplacement du robot. Le dernier bouton « CROIX » permet d'effacer les programmes initialement inscrits dans le robot. Cette plateforme pose différents problèmes pour les élèves.



Figure 5 - le robot Bluebot

Les boutons « TOURNER » sont souvent source d'erreurs chez les élèves qui les programment pour tourner puis avancer directement. Hors ce bouton permet seulement au robot de faire une rotation de 90° sans avancer. Lors de la programmation sur quadrillage, alors qu'un élève avait utilisé ce bouton de manière erronée, un autre élève qui avait suivi son programme s'est empressé d'expliquer ce qu'il venait de se produire : « *y a des personnes qui pensait qu'il avançait quand il tournait alors que non en fait il n'avance pas en même temps, il tourne juste* ».



En observant cette photo, on analyse très clairement le rôle des flèches pour cet élève. L'objectif de cet exercice était d'écrire un mot donné (ici « les ») en réalisant un programme permettant au robot de passer par les différentes lettres de ce mot dans l'ordre et sans passer par aucune autre. L'élève a ici bien utilisé la première flèche pour faire avancer le robot mais il utilise ensuite les flèches en fonction du quadrillage, sans prêter attention au langage de programmation utilisé par le robot. De plus, l'élève écrit ici un nombre de flèches correspondant au nombre de cases et non au déplacement réalisé. L'élève ne semble pas comprendre l'action des flèches. Lors de la réalisation de ce programme sur le robot, celui-ci se rend compte qu'il n'atteint pas l'objectif fixé. Alors que l'utilisation du robot permet d'observer cette erreur et la comprendre, la programmation sur papier ne permet pas cette observation. L'intérêt de la manipulation du robot est ici très important pour permettre aux élèves de construire leurs apprentissages. Cependant il est nécessaire de garder une trace de son programme pour pouvoir le modifier, comme l'a fait remarquer un élève : *« en fait faudrait qu'on ait une feuille pour réécrire ce qu'on dit »*. Cette intervention montre la capacité d'évaluation de l'élève.

Le bouton « PAUSE » questionne beaucoup les élèves sur son utilité comme l'a démontré Spach (2018) dans ses recherches. En effet, dès le début celui-ci a été interprété comme une manière d'arrêter le robot lorsqu'il ne fait pas ce qui est attendu de l'élève, de le mettre en « pause ». Ils ont donc testé leur proposition qui s'est avérée erronée. De la même manière, le bouton « CROIX » est également source de questionnement pour les élèves. Il est difficile pour eux de concevoir qu'un programme écrit est mémorisé par le robot et que chaque instruction donnée en plus s'incrémente dans le programme d'origine. Voici un échange enregistré et retranscrit entre plusieurs élèves sur le rôle de ces deux boutons. Un élève vient de programmer le robot à la suite d'autres élèves, sans avoir effacé les programmes précédents puisqu'il s'agit de la découverte de l'utilité de la « croix ».

« Q. « Mais qu'est-ce qu'il me fait ? »

L. Bah oui t'as pas fait...

Q. J'ai pas réussi, ah bah ça c'est bizarre

L. Il t'écoute pas

Q. Il écoute jamais

Q. Nouvel ordre.

C. Il fait toujours les mêmes choses c'est bizarre.

Q. Attend on va lui mettre la croix, ça fait quoi croix, ça l'arrête ?

L. Pourquoi il s'arrête pas ?

M. Il a fait n'importe quoi du coup le robot à fait n'importe quoi. (En parlant de l'élève qui vient de programmer le déplacement)

Q. Normalement c'est ça, mais pourquoi il fait ça ?

M. Il fait tout à l'envers.

L. Mais on a fait tous les trucs possibles.

L. Fais pause pour l'arrêter.

Q. Faut appuyer sur la croix pour le bloquer.

L. Non ça marche pas on a essayé.

Q. Ah oui faut appuyer sur les deux traits. »

Si l'on reprend les registres de Grugier, les élèves sont au départ dans un registre de participation puisqu'ils jouent avec le robot sans comprendre le rôle de leur action sur celui-ci : « *mais qu'est-ce qu'il me fait ?* ». Au cours de l'échange, les élèves sont en recherche, ils émettent des hypothèses sur la manière dont fonctionne le robot et sur les possibilités qu'il offre. Les élèves travaillent ensemble pour atteindre le but qu'il leur est fixé ou autrement dit le problème posé. Grâce aux différentes actions qu'ils réalisent, ils comprennent qu'ils ont fait une erreur et remettent en question d'abord le robot avant d'essayer de modifier leur programme. Les élèves entrent alors dans un registre d'interprétation pour comprendre le fonctionnement du robot en émettant des hypothèses sur son fonctionnement. On note bien ici la place de l'erreur dans le développement des compétences des élèves qui avancent pas à pas en construisant leurs connaissances ensemble. On peut noter la pertinence des interventions de Q. qui est en réflexion permanente de manière générale. C'est un élève qui est en demande d'apprentissage et qui a de grandes capacités. Cet échange nous montre également la différence entre des élèves qui ont plus de difficulté et qui n'entrent

pas dans une démarche de recherche face à une erreur mais restent bloqués dans la démarche utilisée.

- Le repérage dans l'espace

La question du repérage dans l'espace est très importante lors de la programmation du robot puisque la latéralisation des élèves permet la latéralisation du robot. En effet lorsque les élèves doivent faire tourner le robot pour lui faire atteindre son but, il est nécessaire qu'ils sachent dans quel sens le robot doit tourner. Il est possible d'aider les élèves non latéralisés pour ne pas les pénaliser en programmation en leur faisant une croix sur la main droite par exemple ou avec des bracelets de couleur.

De nombreuses erreurs interviennent lors de la rotation du robot puisqu'il est difficile pour certains élèves de se décentrer. En observant le sens de rotation certains sont capables de se corriger en modifiant celui-ci. On observe sur la copie d'écran suivante d'une vidéo réalisée en classe, le choix d'un élève pour comprendre la rotation du robot et le programmer. Cet élève a donc décidé de se tourner pour expliquer aux autres élèves que le robot n'était pas forcément latéralisé de la même manière que nous tout au long de son déplacement.



Figure 7 - latéralisation du robot

- La position du robot

La position du robot joue un rôle déterminant dans la programmation. Lorsque le robot part devant l'élève, soit dans le même sens que celui-ci, il est plus facile pour l'élève de le faire avancer. Lors du déplacement du robot, celui-ci est amené à tourner et donc à ne plus être dans le même sens que l'élève. Il est donc intéressant d'observer leur manière de programmer la rotation du robot. Certains élèves se tournent en même temps que le robot pour comprendre sa rotation « je me tourne donc maintenant il doit aller à droite » (A.) et savoir quand celui doit avancer ou dans quel sens doit-il tourner. D'autres élèves utilisent leur main comme repère pour imiter le robot et le faire avancer. D'autres encore utilisent la pointe du crayon pour comprendre la rotation du robot et suivre son déplacement pas à pas. Sur l'image ci-dessous nous pouvons observer trois élèves décodant un programme déjà écrit pour trouver le mot codé.

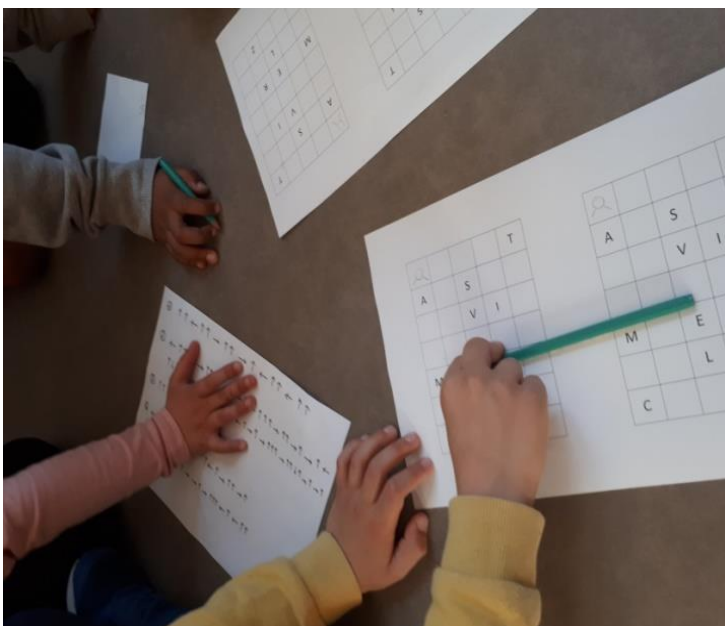


Figure 7 - Stratégies pour décoder un programme : le crayon

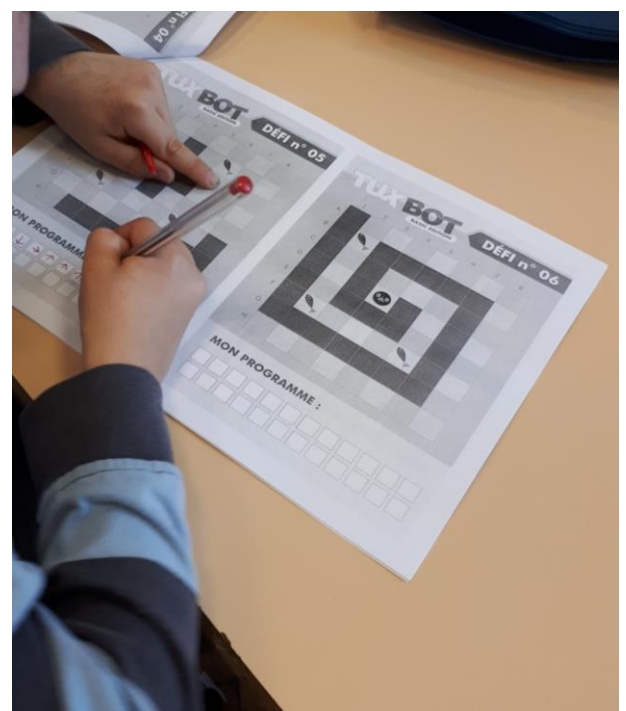


Figure 8 - Stratégies pour coder un programme : le doigt

- Le travail de groupe

Le travail de groupe a été déterminant dans le développement des compétences, notamment pour les élèves en difficulté. Les élèves travaillaient par deux pour créer le programme sur quadrillage avant de l'évaluer sur le robot. Une fois le programme écrit, un élève dictait le programme à l'autre pour que ce deuxième programme le robot. Les compétences langagières ont alors été travaillées sur ces moments de partage. Une fois le programme évalué, les élèves retournaient à leur place pour le modifier, en fonction de ce qu'ils venaient d'observer. Les élèves qui attendaient pour tester leur production participaient à l'évaluation du programme des autres groupes en observant l'écriture de leur algorithme et le retour de l'erreur sur celui-ci. De nombreux échanges formateurs ont donc eu lieu sur ces moments d'évaluation que je transcrirais dans la suite de ce mémoire.

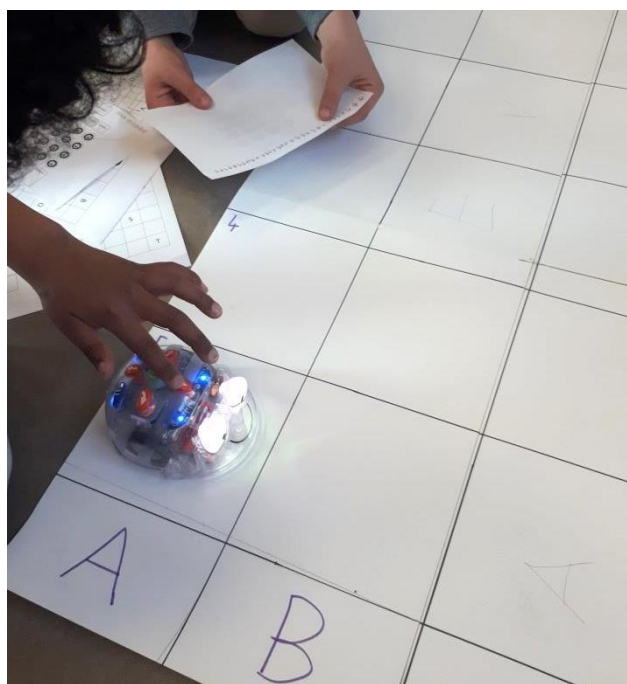


Figure 9 - Deux élèves programment le robot

4.2 L'application Tuxbot

L'application Tuxbot utilise le même langage de programmation que le robot Blue-bot, sa plateforme étant identique. C'est dans ce sens que j'ai choisi de travailler avec ces deux supports pour permettre aux élèves de développer des compétences d'anticipation et d'évaluation au travers des activités branchées.

L'exécution de l'application Tuxbot consiste à programmer le déplacement d'un pingouin pour récupérer tous les poissons situés dans le quadrillage. Le programme ainsi écrit est gardé en mémoire sur l'écran, l'élève peut alors suivre le déplacement du robot en même temps que le programme qu'il a écrit. Il s'agit ici pour moi d'observer le retour de l'erreur sur cette application dans l'apprentissage des élèves. En effet, une fois le programme terminé, il n'est pas possible de revenir sur celui-ci du au paramètre utilisé : le programme est automatiquement réinitialisé. Le paramétrage de l'application joue un rôle important dans la productivité de l'erreur et le retour qu'il renvoie. Deux options sont proposées : « conserver la séquence d'instruction » et « ne pas remettre à zéro le parcours ». Pour éviter la création pas à pas d'un programme et bien développer la pensée informatique des élèves comme l'énonçait J. Wing, il est important que le robot recommence le programme du début à chaque erreur dans le programme.

Les élèves travaillent par groupe de deux ou trois pour réaliser la construction des programmes. Lorsqu'un niveau est terminé, ils passent au suivant, ceux-ci respectant un niveau de difficulté croissant. Les élèves sont donc obligés de réussir le niveau et ainsi modifier leur programme erroné pour avancer. Plusieurs stratégies se mettent en place pour réussir à programmer ce pingouin. Certains élèves ont fait le choix d'écrire leur programme sur une feuille de papier pour le garder en mémoire et pouvoir ainsi revenir dessus, le paramétrage du logiciel ne permettant pas de revenir sur l'erreur puisque le programme n'est pas sauvegardé à la fin de son test. En associant le déplacement du robot et le programme écrit, les élèves peuvent ainsi détecter leur erreur de conception et la corriger.



Figure 10 - Programmation sur Tuxbot et quadrillage papier

Sur la photo ci-dessus, on observe une élève programmant sur l'application Tuxbot. Celle-ci s'aide du quadrillage papier pour écrire son programme et ainsi revenir sur son erreur lors du déplacement du robot sur l'ordinateur. L'utilisation des deux outils facilite le retour sur l'erreur qui semble être plus difficile dans la programmation sur papier. L'application Tuxbot dispose d'une option permettant d'avoir un retour sur l'erreur grâce à la sauvegarde du programme. En effet, il est possible de revenir sur l'écriture de son programme une fois celui-ci testé par une exécution pas à pas avec retour de la machine sur la réussite ou non du programme. Cependant, la modification du programme ne peut se faire que depuis la fin de celui-ci. Le retour productif de l'erreur est donc rendu possible qu'une fois la fonction « conserver la séquence d'instruction » activée. Dans le mode dans lequel on travail, on ne peut avoir de retour sur le programme exécuté et il est impossible de le modifier d'où l'utilisation du quadrillage papier par cet élève.

J'ai retranscrit ci-dessous l'échange enregistré avec un élève qui s'était trompé sur son programme et qui a réussi à modifier son erreur après avoir suivi son programme.

Retranscription élève 1 :

Elève : « On avait réussi à s'arrêter là (en montrant le quadrillage), on a redessiné tout sauf qu'on s'était trompé et du coup on en a oublié un et du coup on n'a pas réussi à aller tout au bout. »

Enseignant : « Et vous les aviez dessiné où ? »

Elève : « Sur notre feuille. »

Enseignant : « Et pourquoi vous ne l'avez pas noté directement sur votre feuille de programme ? Vous n'étiez pas sûrs de votre programme ? »

Elève : « Si on était sûr mais comme il fallait qu'on s'en rappelle on l'a noté à côté comme j'avais déjà commencé à dessiner des flèches sur le programme. »

L'élève a changé de registre, il est maintenant en capacité de modifier son programme pour résoudre l'erreur : il entre dans le registre de modification selon Grugier. L'erreur a du sens pour lui et son camarade de travail et est productive dans l'apprentissage de la programmation. Les élèves écrivaient simultanément le programme sur une feuille et sur l'ordinateur, tout en imaginant son déplacement avant de lancer le robot. Ils ont ensuite suivi le programme qu'ils avaient réalisés et se sont rendu compte qu'il leur manquait une flèche pour arriver jusqu'au bout. Ils ont donc immédiatement modifié leur programme sur leur feuille de programmation. L'application leur a permis de se rendre compte de leur erreur et d'incrémenter une partie du programme à l'ancien sans tout reprogrammer depuis le début.

Au cours de la réalisation de ma séquence, j'ai pu remarquer que certains élèves ne faisaient pas sens à l'erreur produite et reprenaient tout leur programme depuis le début. En observant les élèves, je remarque que les élèves ayant une facilité d'abstraction, notamment en mathématiques, ont beaucoup plus de facilités à modifier leur programme que les autres. Pour les autres, l'impossibilité de revenir sur le programme écrit les empêche de le modifier et ils sont obligés de le reprendre depuis le début. Il aurait été intéressant de modifier les paramètres de l'application qui permettent de revenir sur le programme sans le modifier depuis le début sur l'ordinateur mais le manque de temps ne me l'a pas permis.

4.3 La productivité du retour sur l'erreur

Le retour sur l'erreur semble être interprété différemment selon les élèves, quel que soit l'outil utilisé. Tandis que certains élèves sont capables de comprendre que le robot réalise ce qui lui est demandé et que l'erreur vient donc du programme donné par l'élève, d'autres semblent penser que l'erreur vient du robot. Les réactions des élèves face à l'erreur sont donc très différentes. Tous les élèves ne prennent pas d'information sur l'endroit où ils se sont trompés et ne peuvent donc pas facilement incrémenter un programme à l'initial mais le réécrivent généralement dans sa totalité. Ils ne mettent pas tous de sens. Les élèves atteignent des registres différents, énoncé par Grugier, lors des situations d'enseignements-apprentissage. Dans l'analyse des nombreuses situations suivantes, nous analyserons les registres dans lesquels se situent les élèves.

Les photos suivantes montrent bien la place importante qu'occupe l'erreur en programmation et la nécessité de revenir sur son programme. L'exercice était ici de programmer un robot pour écrire un mot donné. Une fois le programme écrit, les élèves venaient le tester sur le robot pour le vérifier. En analysant le déplacement du robot et en modifiant plusieurs fois leur programme, les élèves ont finalement réussi à trouver la solution.

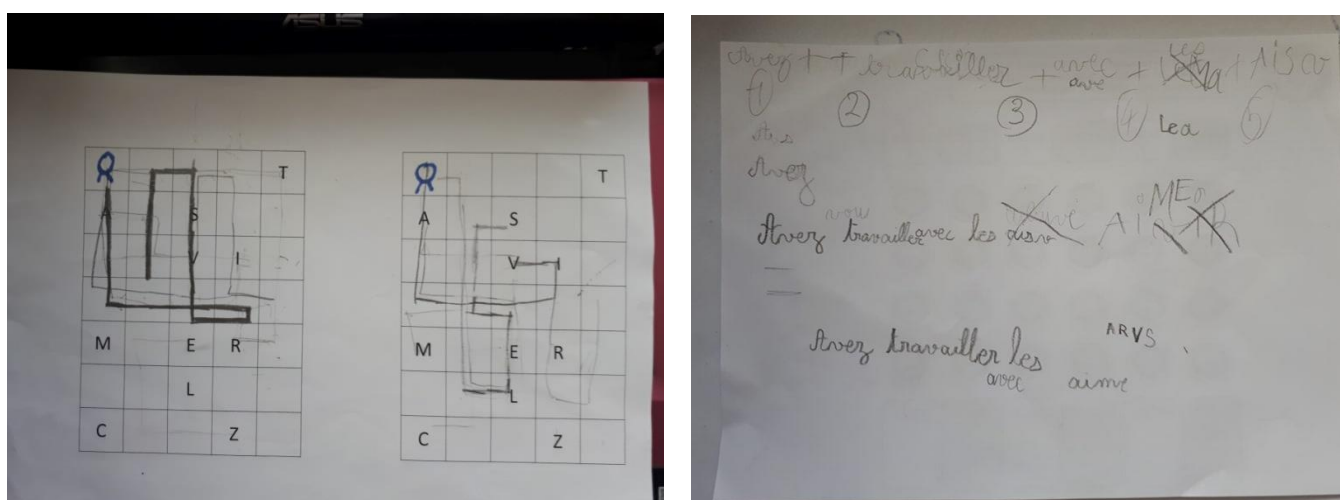


Figure 10 et 11 - Productions d'élèves mettant en avant les erreurs considérées comme productives, dans une situation de décodage.

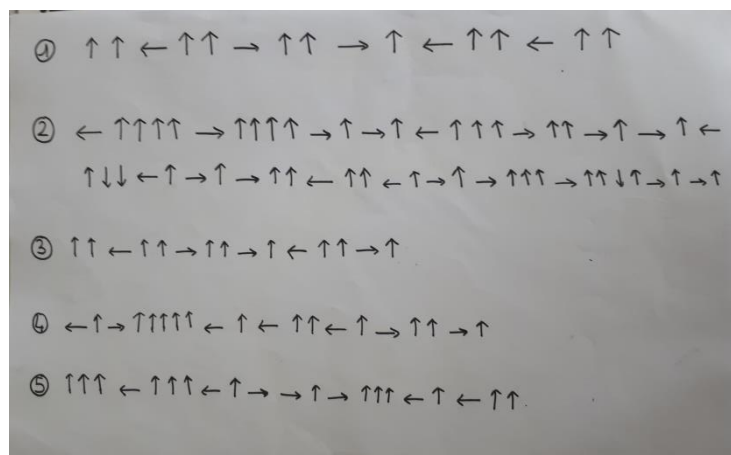


Figure 12 - Codage proposé aux élèves pour retrouver les mots secrets

Les trois situations ci-après montrent la difficulté pour certains de rendre l'erreur productive tandis que pour d'autres elles semblent être un point d'appui.

Situation 1 :

« F. Oh ils ont oublié un tout droit, c'est parce qu'il a dit trois fois c'est pour ça qu'il est parti là. »

L. C'est qu'ils n'avaient pas enlevé la croix

Enseignant : Est-ce que vous savez ou vous vous êtes trompés ?

MA. Euh non

AN. Bah en fait quand ils sont arrivés sur le « u » ils ont reculé alors qu'ils devaient tourner et avancer. »

Dans cette situation, nous remarquons que le programmeur, ici MA, Ne comprends pas son erreur et n'est pas capable de rendre celle-ci productive puisqu'il ne suit pas le déplacement du robot en même temps que le programme qu'il a écrit. Ceci s'explique par le langage qu'il utilise dans l'écriture de son programme. Il ne met pas de sens dans la rotation du robot et donc le déplacement de celui-ci. Il se situe dans le registre de participation. A l'inverse, les élèves qui observent le programme comprennent que l'élève s'est trompé sur la fin de son programme et cherchent la modification qu'il faudrait effectuer pour que celui-ci soit bon. Ces derniers ont donc atteint le registre de modification.

Situation 2 :

« M. Il est pas arrivé au bout il me manque un tout droit

N. Pourquoi il recule sans arrêt

M. On a oublié d'avancer et tourner

M. Avant de reculer on doit mettre un avancer non trois avancer »

On peut analyser le rôle productif de l'erreur chez le programmeur, ici M, qui analyse le déplacement du robot et essaye d'incrémenter un déplacement dans celui déjà produit sans pour autant vouloir tout modifier.

Situation 3 :

« N. Faut appuyer sur go

Le robot n'arrive pas à la bonne case

Enseignant. *Est-ce que tu as réussi ?*

L. Non mais je sais pourquoi, il fallait appuyer tout droit après.

N. J'ai pas trop compris, comment il fait pour rouler tout seul

E. Bah il comprend

Enseignant. *Pourquoi il continue ?*

E. Ah bah oui pourquoi il continue ? Ah parce qu'elle a fait trop... »

De la même manière dans cette situation, le programmeur identifie tout de suite son erreur et ajoute l'action manquante à la fin de son programme. Les observateurs ne semblent pas comprendre comment le robot avance et comment L. a réussi à programmer son robot. A la fin de son programme, le robot continue d'avancer et un élève relève cette situation anormale, il comprend alors qu'elle a donné trop d'indications et qu'il faut en supprimer.

L'erreur est constructive dans l'acquisition de l'objectif de programmation et a permis une grande marge de progression notamment pour les élèves ayant des facilités de compréhension. La mise en place de cette séquence met en avant les difficultés à comprendre l'enjeu et l'abstraction du robot pour les élèves en difficulté bien que la motivation du robot les emmène également à réaliser les objectifs. Il semblerait donc que selon les élèves, l'erreur est considéré comme un outil qui permet de revenir sur le programme construit tandis que pour d'autre

l'erreur est un outil essai-erreur qui permet de vérifier son programme sans pour autant le modifier.

Les deux méthodes de programmation analysées sont différentes mais complémentaires. Elles permettent toutes deux une analyse de l'erreur dans la programmation qui trouve des points communs. Chacune de ces deux applications est motivante dans la réalisation de la tâche et permet aux élèves de construire ses apprentissages dans une démarche d'investigation. L'erreur tient une place importante dans cette construction du savoir mais ne suffit pas à tous les élèves. Comme l'énonçait Spach, le rôle de l'enseignant est déterminant dans la proposition des situations. Les élèves ayant plus de difficultés dans les apprentissages mathématiques et logiques n'interprètent pas l'erreur pour que celle-ci devienne productive. La situation choisie par l'enseignant doit donc s'adapter aux élèves en présentant des situations complexes pour certains tandis que des situations simples doivent être proposées aux élèves en difficulté pour rester dans la zone proximale de développement des élèves, définie par Vygotsky.

5. Conclusion

L'informatique prend une place de plus en plus importante dans notre société et donc dans l'éducation. Il est important de former des futurs citoyens compétents et responsables face aux nouvelles technologies numériques. Nombreux auteurs tels que Wing et Grugier s'accordent à dire que le développement de la pensée informatique et l'apprentissage de la programmation permettent de développer des compétences transférables à d'autres enseignements, notamment dans la formation d'un esprit logique.

Par ailleurs, l'erreur est un outil qui a repris sa place de formation dans les apprentissages au cours des dernières années. Souvent vécue comme un affront, elle est repensée et intègre totalement les apprentissages. En prenant en compte l'importance de l'erreur, l'élève entre dans une démarche de recherche, d'investigation, lui permettant de comprendre son action et ainsi évaluer le résultat de celle-ci. Il est important de repréciser la place de l'erreur dans les apprentissages comme outil et le numérique est un moyen de le mesurer.

Pour garantir la productivité de l'erreur dans l'apprentissage de la programmation avec l'utilisation du Bluebot, il est important de garder une trace du programme et l'écriture de celui-ci permettra de travailler les compétences d'anticipation et d'évaluation. Seule l'écriture du programme permet la modification et l'évaluation de son action sur le robot. Bien qu'il soit un élément nécessaire dans l'observation de l'erreur, le déplacement du robot ne suffit pas à l'élève pour avoir un retour productif de son erreur s'il ne garde pas trace de son action. De la même manière, l'application Tuxbot nécessite la mise en place de l'option de sauvegarde du programme avec retour du robot au départ pour permettre à l'élève d'observer l'erreur dans son programme. Sans cette option, la capacité d'évaluation de l'élève est rendu difficile. Il est donc important de penser sa séquence et les modalités de sa mise en place. Par ailleurs, le travail en binôme permet aux élèves de confronter leurs points de vue et ainsi développer des compétences plus solides en programmation ou dans d'autres enseignements.

Lors de la conception de la séance, la différenciation doit être pensée comme le montre l'analyse des situations. En effet, les élèves n'ont pas les mêmes capacités d'apprentissage et la différenciation des entrées dans la programmation est un gage de réussite pour tous les élèves. La capacité d'abstraction, nécessaire à l'apprentissage de la programmation, est différente selon les élèves et la proposition de situations simples pour les élèves en difficulté semble primordiale. A l'inverse, pour les élèves les plus à l'aise il est envisageable de proposer des situations plus complexes permettant de développer le maximum de leurs compétences. Il s'agit bien d'évaluer la zone proximale de développement de chaque élève pour garantir le développement des compétences d'anticipation et d'évaluation chez tous les élèves.

L'erreur ne devient donc productive que lorsque l'élève comprend l'importance de son action sur le robot et qu'il entre dans un registre de modification, identifié par Grugier. Il est donc nécessaire de réfléchir à la manière d'atteindre ce registre pour tous les élèves.

Bibliographie

Baron G. L. et Drot-Delange B. (2016), L'éducation à l'informatique à l'école primaire, *Bulletin de la société informatique de France*, (8), 73-79. En ligne : <https://hal.archives-ouvertes.fr/hal-01403598>

Brouze E. (2016). Le numérique à l'école change-t-il vraiment le rapport à l'école ? *L'Obs avec Rue 89*. En ligne : <https://www.nouvelobs.com/rue89/rue89-education/20160427.RUE2766/le-numerique-a-l-ecole-change-t-il-vraiment-le-rapport-a-l-erreur.html>

Conseil national du numérique. (2014). *Jules ferry 3.0, Bâtir une école créative et juste dans un monde numérique*, Rapport du Conseil National du Numérique.

Delarbre, P. (2017). *Initiation à la programmation à l'école primaire : les activités « débranchées »*, (mémoire de certificat à l'aptitude aux fonctions d'instituteur ou de professeur des écoles maître formateur (CAFIPEMF), académie de Lyon). En ligne : ekladata.com/1_omEKfJxnpu5iOldtTcvKJzL8E/Memoire-Peggy-DELABRE-.pdf

Ministère de l'éducation nationale et de la jeunesse, (s.d.). « L'école change avec le numérique », Devenir enseignant. En ligne : <http://www.devenirenseignant.gouv.fr/pid34343/l-ecole-change-avec-numerique.html> [consulté le 03-04-2019]

De Vecchi, G. (2014). *Aider les élèves à apprendre*, Hachette éducation

Dowek, G. (2011). Les quatre concepts de l'informatique. Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques. Actes du quatrième colloque international DIDAPRO 4, Université de Patras.

Grugier, O. (2018). *Rencontres entre des élèves de maternelle et des robots programmables*. Des savoirs technologiques et informatiques, Université Paris Descartes.

Ministère de l'éducation Nationale. (2015). Programme d'enseignement de l'école maternelle. Arrêté du 18-2-2015, Paris.

Ministère de l'éducation Nationale. (2015). Programmes d'enseignement du cycle des apprentissages fondamentaux (cycle 2), du cycle de consolidation (cycle 3) et du cycle des approfondissements (cycle 4), arrêté du 9-11-2015, Paris.

Ministère de l'éducation Nationale. (2015). Socle commun de connaissances, de compétences et de culture, décret n°2015-372 du 31-3-2015, Paris.

Ministère de l'éducation Nationale. (2016). Initiation à la programmation aux cycles 2 et 3, MENESR, Paris.

Nogry, S. (2018). Comment apprennent les élèves au cours d'une séquence de robotique éducative en classe de CP ?, Didapro 7, HAL, Suisse.

Roy, D. (2017). L'informatique et le numérique dans la classe: Qui, quoi, comment ? Namur: Presses Universitaires de Namur.

Spach, M. (2018). Robotique pédagogie à l'école primaire, découverte de la touche « pause », des possibilités intéressantes. *Adjectif.net* Mis en ligne mercredi 20 juin 2018. En ligne : <http://www.adjectifs.net/spip/spip.php?article470>

Tchounikine, P. (2016). *Initier les élèves à la pensée informatique et à la programmation avec scratch*, (Université Grenoble-Alpes). En ligne : <http://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.pdf>.

Wing, J. (2006). *Computational thinking*, Communications of the Association for Computing Machinery. 49(3):33–35. En ligne : <http://www.cs.cmu.edu/afs/cs/usr/wing/www/ct-french.pdf>.

Annexe 1

Fiche de préparation de séquence	
Domaines : Les langages pour penser et communiquer – les méthodes et outils pour apprendre – systèmes naturels et systèmes techniques	Cycle 2 – CE1
Disciplines : Questionner le monde – mathématiques	Période 4
Séquence : programmation de robots blue-bot	Nombre de séances : 7 séances
Objectifs de la séquence : <ul style="list-style-type: none"> - Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran - Se repérer, s'orienter en utilisant des repères - Adopter une démarche scientifique : utilisation d'un langage spécifique, contrôle, essais-erreurs - Développer l'abstraction : apprendre à anticiper l'effet de telle ou telle séquence d'instructions avant même de la faire exécuter 	
Objectifs langagiers : <ul style="list-style-type: none"> - Utiliser le langage informatique et mathématique - Utiliser le vocabulaire spatial appris - Exprimer son point de vue et accepter celui des autres 	
Compétences du socle commun : être capable de <ul style="list-style-type: none"> - Modéliser des situations - Savoirs que des langages informatiques sont utilisés pour programmer des outils numériques - Connaître les principes de base de la conception de programme - Coopérer, réaliser un projet - S'engager dans une démarche de résolution de problème - Analyser et exploiter les erreurs, mettre à l'essai plusieurs solutions - Coopérer et faire preuve de responsabilité vis-à-vis d'autrui - Participer à un projet de groupe, d'engager dans sa mise en œuvre - Mener une démarche d'investigation 	

Séances	Objectifs et déroulement	Matériel
Séance n°1 : Lancement du projet	<p>Objectif : émettre des hypothèses sur le déplacement d'un robot, entrer dans une démarche de projet</p> <p>Présentation de la vidéo d'un robot qui se déplace : recueil des représentations initiales sur le déplacement. Affiche : nous pensons que Annoncer l'objectif final : programmer le déplacement d'un robot.</p>	Vidéo robot Affiche
Séance n°2 : Approche corporelle – robot-idiot	<p>Objectif : donner des ordres précis pour qu'un élève se déplace dans un espace, valider l'hypothèse de la programmation d'un déplacement</p> <p>Rappel de la séance précédente Jeu du robot idiot : déplacement par le corps sur un quadrillage au sol par groupe de 2. Donner un point de départ et un point d'arrivée et les laisser se guider. Introduire le codage écrit par les flèches en empêchant le langage oral. Synthèse : le code est reproductible, stable</p>	Quadrillage au sol
Séance n°3 : Approche corporelle – trouve le bon programme	<p>Objectif : retrouver les ordres précis qui permettent à un élève de réaliser le déplacement attendu</p> <p>Rappel de la séance précédente Les élèves doivent retrouver le programme qui mène au point d'arrivée donné par l'enseignant. Faire une affiche pour expliquer ce que l'on a appris de la programmation (flèche écrite)</p>	Programme Quadrillage
Séance n°4 : phase écrite	<p>Objectif : concevoir un programme pour définir un déplacement complexe d'un personnage en combinant des instructions simples</p> <p>Rappel de la séance précédente Programmer des déplacements sur le quadrillage en observant le sens du robot. Utiliser le code vu lors des séances précédentes pour programmer. Mise en commun, bilan collectif.</p>	Quadrillage papier
Séance n°5 : robot et numérique	<p>Objectif : découvrir le robot et se familiariser avec le déplacement ; réaliser un programme donné</p> <p>Rappel des séances Découverte du robot en collectif : premier essais pour voir comment fonctionne le robot, découverte de la croix. Lire des programmes de déplacements sur quadrillage : se décentrer pour visualiser et décoder un déplacement. Plusieurs programmes donnés, sur quadrillage les élèves essayent de découvrir le mot puis vérification en programmant le robot. (travail en binôme)</p>	Robot / quadrillage

	<p>Les autres élèves travaillent en autonomie sur le plan de travail, échange des ateliers ensuite.</p> <p>Conclusion trace écrite : mode d'emploi du robot</p>	
Séance n°6 : conception de programme	<p>Objectif : concevoir un programme et le vérifier ; programmer les rotations du robot</p> <p>Rappel : un élève reprend le rôle du robot idiot face aux élèves pour permettre d'observer la latéralisation du robot. observation en classe entière, utilisation de « tourne à gauche », « tourne à droite » pour comprendre le sens de rotation.</p> <p>Coder des mots grâce à un programme : conception du programme écrit à l'aide d'un quadrillage puis vérification sur le robot (en binôme).</p> <p>Ateliers tournants avec programmation sur ordinateur (Tuxbot).</p>	Robot Quadrillage/tapis
Séance n°7 : réalisation du programme reçu	<p>Objectif : réaliser un programme donné</p> <p>Rappel des séances précédentes</p> <p>Continuer la séance précédente pour créer des programmes : programmation sur papier puis sur robot en binôme.</p> <p>Ateliers tournants avec l'ordinateur et la programmation sur papier.</p> <p>Niveau de difficulté croissant dans les exercices de programmation.</p> <p>Conclusion</p>	Robot Programme
Prolongements : réinvestissement dans les plans de travail		

Résumé en français

Ce mémoire présente les résultats d'une recherche visant à étudier la place de l'erreur dans le domaine informatique, plus particulièrement en programmation, dans une classe de CE1. En invoquant des recherches récentes sur les processus d'apprentissages (Grugier et Spach) et en analysant une situation d'apprentissage en classe, il s'agit de comprendre quelle place tient l'erreur dans la réalisation des objectifs et la productivité qu'elle implique pour les élèves. Au-delà de la programmation il convient de comprendre quel rôle tient l'erreur lorsqu'elle est induite par un outil et non par l'enseignant.

Mots-clés : programmation, robotique, erreur, enseignement primaire, didactique de l'informatique.

Résumé en espagnol

Este trabajo presente los resultados de una investigación centrándose en la función que tiene el error en el ámbito informático, en particular en la programación, con niños de primaria. Invocando investigaciones recientes sobre los procesos de aprendizaje (Grugier y Spach) y analizando una situación de aprendizaje en el aula, se trata de comprender qué lugar ocupa el error en la consecución de los objetivos y la productividad que ella implica para los alumnos. Más allá de la programación, es preciso comprender el papel que desempeña el error cuando es inducido por una herramienta y no por el profesor.

Mots-clés : programación, error, educación primaria, didáctica de la informática.