
ÿ Analyse de l'impact de la mise en Suvre de scénarios d'étayage visant le développement de la pensée informatique et algorithmique dans l'enseignement fondamental

Auteur : Marquet, Pascale

Promoteur(s) : Denis, Brigitte

Faculté : ÿ Faculté de Psychologie, Logopédie et Sciences de l'Education

Diplôme : Master en sciences de l'éducation, à finalité spécialisée en enseignement

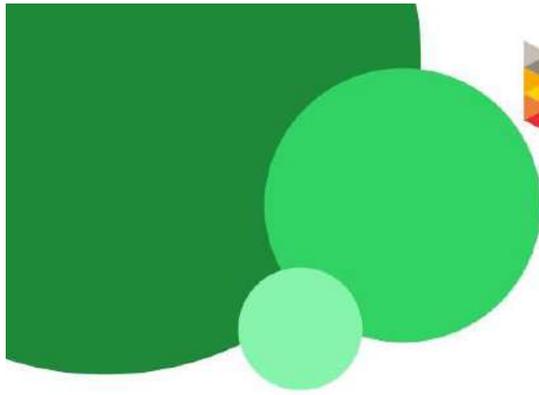
Année académique : 2021-2022

URI/URL : <http://hdl.handle.net/2268.2/16157>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.



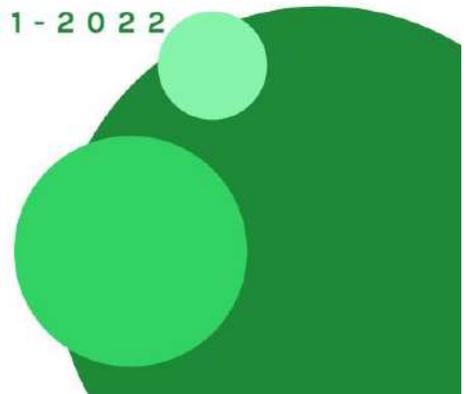
LIÈGE université
Psychologie, Logopédie
& Sciences de l'Éducation

Analyse de l'impact de la mise
en oeuvre de scénarios
pédagogiques et d'outils
d'étayage visant le
développement de la pensée
informatique et algorithmique
dans l'enseignement fondamental

Mémoire présenté par
Pascale MARQUET
en vue de l'obtention du grade de
Master en Sciences de l'Éducation

Année académique 2021-2022

Promotrice: Mme Brigitte Denis
Lecteurs: Mme Margault Sacre
& Mr Robert Reuter



Remerciements

Je tiens à adresser mes plus sincères remerciements à celles et ceux qui ont contribué à l'élaboration de ce mémoire et plus largement à la réalisation de ce master.

Tout d'abord, je tiens à remercier Madame Denis, ma promotrice, pour son aide, ses précieux conseils et sa disponibilité tout au long de ces études.

Je remercie aussi Madame Higuët et Madame Joris, ses assistantes, pour leur soutien et leurs conseils judicieux.

Merci également à mes lecteurs, Madame Sacre et Monsieur Reuter, pour l'intérêt porté à ce mémoire.

En outre, je tiens à remercier ma direction, mes collègues et tous mes petits élèves pour leur soutien et leur motivation tout au long de ce travail.

Enfin, mes dernières pensées se dirigent vers mes proches, mes amis et ma famille qui ont su faire preuve de patience et de soutien dans les moments les plus difficiles afin de me donner le courage et la force d'en être arrivée là aujourd'hui. Merci à ma copine de galère, Laurence, sans qui je n'aurais jamais commencé cette aventure. C'est aussi grâce à toi que j'ai grandi tout au long de ces quatre dernières années.

Merci à toi Olivier d'avoir géré le quotidien et d'avoir été un pilier sur lequel j'ai pu me reposer.

Merci à ma fille, Victoria, d'avoir partagé sa maman. J'espère que cette expérience pourra te permettre de comprendre qu'avec de la motivation, du courage et du travail on peut entreprendre de réaliser n'importe quel rêve.

Tables des matières

1.	Introduction	11
2.	Revue de la littérature	14
2.1.	Qu'est-ce que la pensée informatique ?	14
2.2.	Quelles compétences sont développées en pensée informatique et algorithmique ?	16
2.2.1.	Les pays anglo-saxons.....	16
2.2.2.	L'Australie.....	17
2.2.3.	L'Europe.....	18
2.3.	Quelles approches sont utilisées pour développer les compétences en pensée informatique ?	19
2.3.1.	L'informatique débranchée	20
2.3.2.	Les activités de programmation visuelle	21
2.3.3.	Les activités de robotique pédagogique dites « branchées ».....	21
2.4.	Quelles sont les stratégies de programmation mises en œuvre par les apprenants ?	23
2.5.	Comment évaluer les compétences en pensée informatique et algorithmique ?.....	24
2.6.	Quels sont les enjeux et les défis relatifs à la pensée informatique ?.....	27
2.6.1.	Les stratégies d'apprentissage pour encourager le développement de la pensée informatique en fonction du genre	27
2.6.2.	L'usage de techniques d'étayage.....	30
2.6.2.1.	L'apprentissage collaboratif.....	31
2.6.2.2.	L'usage de techniques d'étayages et de la robotique pédagogique.	32
2.6.3.	Quelques pistes pour l'enseignement de la pensée informatique.....	36
3.	Question de recherche et hypothèses.....	37
4.	Méthodologie	38
4.1.	Description du contexte et de la population échantillonnée.....	38
4.2.	Vigilances éthiques	39

4.3.	Schéma des différentes étapes de notre dispositif	40
4.4.	Conception d'un scénario pédagogique incluant deux techniques d'étayage	40
4.4.1.	Les techniques d'étayage	40
4.4.1.1.	Première technique d'étayage (E1)	41
4.4.2.	Le scénario pédagogique	43
4.4.2.1.	Les activités débranchées	44
a)	Première séance	44
b)	Deuxième séance	44
4.4.2.2.	Les activités branchées	45
a)	Troisième séance	46
b)	Quatrième séance	47
c)	Cinquième séance	47
d)	Sixième séance	48
e)	Septième séance	48
4.5.	Recueil et analyse de données	48
4.5.1.	L'évaluation des compétences en PIA	49
4.5.1.1.	Pré-test, post-test immédiat et post-test différé	49
a)	Conception du test	49
b)	Calibrage du test	51
c)	Passation du test	52
d)	Analyse des données	54
4.5.1.2.	La grille d'observation	54
4.5.1.3.	Analyses statistiques	56
4.6.	Limites méthodologiques	57
5.	Présentation des résultats et interprétation	58
5.1.	Analyse de la réussite des items de l'évaluation	58

5.2.	Acquisition des compétences en PIA (H1)	60
5.2.1.	Résultats de tous les élèves aux pré-test, post-test immédiat et post-test différé	60
5.2.2.	Analyse de l'effet d'apprentissage	61
5.2.3.	Analyse du coefficient de variation.....	61
5.2.5.	Analyse des erreurs commises par les élèves présentant une perte relative entre le pré-test et le post-test immédiat.	65
5.2.5.1.	Élève n°2.....	66
a)	Analyse de l'item échoué au pré-test.....	66
b)	Analyse des items échoués au post-test immédiat.....	67
c)	Analyse de l'item échoué au post-test différé	68
d)	Analyse des abstentions.....	68
e)	Liens entre les conduites témoins et les compétences	69
5.2.5.2.	Élève n°11	69
a)	Analyse des items échoués	70
b)	Liens entre les conduites témoins et les compétences	71
5.2.5.3.	Élève n°14.....	71
a)	Analyse des items échoués aux post-tests immédiat et différé.....	72
b)	Liens entre les conduites témoins et les compétences	73
5.3.	Maintien des compétences acquises dans le temps (H2).....	73
5.4.	Acquisition des compétences en PIA selon la technique d'étayage (H3).....	75
5.4.1.	Résultats des élèves ayant bénéficié de la première technique d'étayage : utilisation des cartes de programmation.....	75
5.4.2.	Résultats des élèves ayant bénéficié de la seconde technique d'étayage : l'écriture	76
5.4.3.	Analyse des effets d'apprentissage en fonction de la technique d'étayage reçue	77
5.4.4.	Analyse des occurrences liées à la grille d'observation.....	77

5.5.	Acquisition des compétences en PIA en fonction de la technique d'étayage et du genre (H4)	78
5.5.1.	Résultats des filles dans les deux conditions d'étayage	78
5.5.2.	Analyse des occurrences liées à la grille d'observation chez les filles	79
5.5.3.	Résultats des garçons dans les deux conditions d'étayage.....	79
5.5.4.	Analyse des occurrences liées à la grille d'observation chez les garçons.....	80
6.	Discussion	81
6.1.	L'évaluation	81
6.2.	Acquisition des compétences en PIA (H1)	82
6.3.	Maintien des compétences acquises dans le temps (H2).....	83
6.4.	Acquisition des compétences en PIA selon la technique d'étayage (H3).....	83
6.5.	Acquisition des compétences en PIA en fonction de la technique d'étayage et du genre (H4)	84
6.6.	Perspectives	85
7.	Conclusion.....	87
8.	Bibliographie	89

Table des illustrations

FIGURE 1 / INTERACTIONS EN JEU (DENIS, 2020, P.7)	16
FIGURE 2 / EXEMPLE DE QUESTION.....	26
FIGURE 3 / A FIVE-STEP RESEARCH PLAN FOR CT EDUCATION (ANGELI & GIANNAKOS, P.2, 2020).....	36
FIGURE 4 / RÉPARTITION DE L'ÉCHANTILLON.....	39
FIGURE 5 / DIFFÉRENTES ÉTAPES DU DISPOSITIF	40
FIGURE 6 / FICHES AU FORMAT A5.....	41
FIGURE 7 / CARTES DE PROGRAMMATION PRÉSENTES DANS LE KIT "LEARNING RESOURCES"	41
FIGURE 8 / FLÈCHE "AVANCE"	42
FIGURE 9 / FLÈCHE "AVANCE" ACCOMPAGNÉE DE LA REPRÉSENTATION DE LA SOURIS-ROBOT	42
FIGURE 10 / REPRÉSENTATION DU DÉFI À RÉALISER.....	42
FIGURE 11 / FICHE DÉFI	42
FIGURE 12 / EXEMPLE DE FICHE ACTIVITÉ 3	45
FIGURE 13 / FLÈCHES FORMAT A5	45
FIGURE 14 / ROBOT-SOURIS CODE & GO	45
FIGURE 15 / KIT D'ACTIVITÉS « LEARNING RESOURCES »	45
FIGURE 16 / CARTES DE PROGRAMMATION X2 / X3	46
FIGURE 17 / REPRÉSENTATION DE LA SOURIS.....	46
FIGURE 18 / ÉTIQUETTES FONCTIONS.....	47
FIGURE 19 / CARTE D'ACTIVITÉ.....	47
FIGURE 20 / FICHE D'ACTIVITÉ N°5 ACCOMPAGNÉE DES COMMANDES.....	48
FIGURE 21 / EXEMPLE D'ITEM (1)	51
FIGURE 22 / EXEMPLE D'ITEM (2)	51
FIGURE 23 / EXPLICATION DES COMMANDES.....	53
FIGURE 24 / EXEMPLE DE QUESTION ANIMÉE.....	53
FIGURE 25 / EXTRAIT DE LA GRILLE D'OBSERVATION.....	55
FIGURE 26 / EXEMPLE DE CODAGE DE LA GRILLE D'OBSERVATION.....	56
FIGURE 27 / POURCENTAGE MOYEN DE RÉUSSITE POUR CHAQUE ITEM	58
FIGURE 28 / ITEM N°15.....	58
FIGURE 29 / ITEM N°11.....	59
FIGURE 30 / ITEM N°12.....	59
FIGURE 31 / POURCENTAGE DE RÉUSSITE EN FONCTION DES SOUS-COMPÉTENCES VISÉES	59
FIGURE 32 / POURCENTAGE DES CONDUITES OBSERVÉES LIÉES À CHAQUE COMPÉTENCE CIBLE. N=983.....	63
FIGURE 33 / DÉTAIL DU POURCENTAGE DES CONDUITES OBSERVÉES LIÉES À CHAQUE SOUS-COMPÉTENCE. N=983.....	64
FIGURE 34 / POURCENTAGE DES CONDUITES OBSERVÉES EN FONCTION DES COMPORTEMENTS POUR CHAQUE COMPÉTENCE & SOLLICITATIONS DE L'ENSEIGNANTE	65
FIGURE 35 / RÉPONSE D'E2 À L'ITEM N°9 AU PRÉ-TEST	66

FIGURE 36 / RÉPONSE D'E2 À L'ITEM N°21 AU POST-TEST IMMÉDIAT.....	67
FIGURE 37 / RÉPONSE D'E2 À L'ITEM N°20 AU POST-TEST IMMÉDIAT.....	67
FIGURE 38 / RÉPONSE D'E2 À L'ITEM N°22 AU POST-TEST IMMÉDIAT.....	67
FIGURE 39 / RÉPONSE D'E2 À L'ITEM N°23 AU POST-TEST IMMÉDIAT.....	68
FIGURE 40 / RÉPONSE D'E2 À L'ITEM N°21 AU POST-TEST DIFFÉRÉ.....	68
FIGURE 41 / ITEM N°15.....	68
FIGURE 42 / ITEM N°18.....	69
FIGURE 43 / RÉPONSE D'E11 À L'ITEM N°20 AU POST-TEST IMMÉDIAT.....	70
FIGURE 44 / RÉPONSE D'E11 À L'ITEM N°21 AU POST-TEST IMMÉDIAT.....	70
FIGURE 45 / RÉPONSE D'E11 À L'ITEM N°22 AU POST-TEST IMMÉDIAT.....	70
FIGURE 46 / RÉPONSE D'E11 À L'ITEM N°23 AU POST-TEST IMMÉDIAT.....	71
FIGURE 47 / POURCENTAGES DES CONDUITES TÉMOINS DE L'ACQUISITION DES SOUS-COMPÉTENCES / E11. N=54.....	71
FIGURE 48 / RÉPONSE D'E14 À L'ITEM N°9 AU POST-TEST IMMÉDIAT.....	72
FIGURE 49 / RÉPONSE D'E14 À L'ITEM N°15 AU POST-TEST DIFFÉRÉ.....	72
FIGURE 50 / RÉPONSE D'E14 À L'ITEM N°23 AU POST-TEST DIFFÉRÉ.....	72
FIGURE 51 / POURCENTAGES DES CONDUITES TÉMOINS DE L'ACQUISITION DES SOUS-COMPÉTENCES / E14. N=54.....	73
FIGURE 52 / GAINS RELATIFS MOYENS POUR TOUS LES ÉLÈVES.....	74
FIGURE 53 / ÉVOLUTION DES MOYENNE AUX DIFFÉRENTS TESTS EN PREMIÈRE CONDITION D'ÉTAYAGE.....	75
FIGURE 54 / ÉVOLUTION DES MOYENNES AUX DIFFÉRENTS TESTS EN SECONDE CONDITION D'ÉTAYAGE.....	76
FIGURE 55 / GAIN RELATIF MOYEN EN FONCTION DE LA TECHNIQUE D'ÉTAYAGE.....	77
FIGURE 56 / POURCENTAGES DES COMPORTEMENTS OBSERVÉS POUR CHAQUE COMPÉTENCE EN FONCTION DE LA TECHNIQUE D'ÉTAYAGE.....	77
FIGURE 57 / GAINS RELATIFS MOYENS DES FILLES SUIVANT L'ÉTAYAGE.....	78
FIGURE 58 / POURCENTAGE DES CONDUITES OBSERVÉES LIÉES À CHAQUE COMPÉTENCE CIBLE EN FONCTION DE LA TECHNIQUE D'ÉTAYAGE CHEZ LES FILLES. ÉTAYAGE ÉCRITURE : N=229 ; ÉTAYAGE CARTES : N=320.....	79
FIGURE 59 / GAINS RELATIFS MOYENS DES GARÇONS EN FONCTION DE L'ÉTAYAGE.....	79
FIGURE 60 / POURCENTAGE DES CONDUITES OBSERVÉES LIÉES À CHAQUE COMPÉTENCE CIBLE EN FONCTION DE LA TECHNIQUE D'ÉTAYAGE CHEZ LES GARÇONS. ÉTAYAGE ÉCRITURE : N=271 ; ÉTAYAGE CARTES : N=163.....	80

TABLEAU 1 / SOUS-COMPÉTENCES CIBLES EN FONCTION DES SÉANCES	43
TABLEAU 2 / TABLEAU RÉCAPITULATIF DES DIFFÉRENTS ITEMS DE NOTRE ÉVALUATION	50
TABLEAU 3 / RÉSULTATS DE TOUS LES ÉLÈVES AUX TESTS	60
TABLEAU 4 / COEFFICIENT DE VARIATION DES PRÉ-TEST ET POST-TEST IMMÉDIAT	62
TABLEAU 5 / POURCENTAGE DES CONDUITES OBSERVÉES POUR CHAQUE ÉLÈVE AU REGARD DE CHAQUE COMPÉTENCE ET SOUS-COMPÉTENCE TRAVAILLÉE	62
TABLEAU 6 / POURCENTAGES DES INTERVENTIONS DE L'ENSEIGNANTE (VERBALES OU DIRECTEMENT EN LIEN AVEC LE ROBOT PÉDAGOGIQUE) POUR CHAQUE COMPÉTENCE ET SOUS-COMPÉTENCE TRAVAILLÉE	63
TABLEAU 7 / SCORES OBTENUS PAR E2 AUX DIFFÉRENTS TESTS	66
TABLEAU 8 / ITEMS ÉCHOUÉS OU ABSTENUS PAR E2 AUX DIFFÉRENTS TESTS	66
TABLEAU 9 / SCORES OBTENUS PAR E11 AUX DIFFÉRENTS TESTS	69
TABLEAU 10 / ITEMS ÉCHOUÉS OU ABSTENUS PAR E11 AUX DIFFÉRENTS TESTS	70
TABLEAU 11 / SCORES OBTENUS PAR E14 AUX DIFFÉRENTS TESTS	71
TABLEAU 12 / ITEMS ÉCHOUÉS OU ABSTENUS PAR E14 AUX DIFFÉRENTS TESTS	72
TABLEAU 13 / RÉSULTATS DES ÉLÈVES EN PREMIÈRE CONDITION D'ÉTAYAGE	75
TABLEAU 14 / RÉSULTATS DES ÉLÈVES EN SECONDE CONDITION D'ÉTAYAGE	76
TABLEAU 15 / RÉSULTATS DES FILLES DANS LES DEUX CONDITIONS D'ÉTAYAGE	78
TABLEAU 16 / RÉSULTATS DES GARÇONS DANS LES DEUX CONDITIONS D'ÉTAYAGE	79

1. Introduction

« Everybody should learn to program a computer, because it teaches you how to THINK » (Jobs, 1995).

Que retenir de cela ? Cette vision de Steve Jobs nous amène à considérer pourquoi nous devrions apprendre aux enfants à coder : cela leur permettrait d'obtenir le reflet de leur processus de pensée. L'objectif de la programmation n'est pas d'obtenir de futurs informaticiens capables de créer le dernier réseau social à la mode ou la toute nouvelle application révolutionnaire, même si, certains choisiront cette voie.

Dans ce cadre, l'école se trouve inévitablement concernée par cette thématique. Introduire des activités de programmation en classe ou en dehors de celle-ci ne date pas d'hier (Papert, 1980), même si ce slogan est en quelque sorte teinté d'urgence dans une société qui tend à faire acquérir les compétences du XXI^e siècle.

L'enseignement de la programmation poursuit deux objectifs différents, mais pas nécessairement opposés. Tantôt employé comme un but en soi, tantôt comme un moyen d'apprendre autre chose, il peut également être inclus dans un cadre moins circonscrit et parler de culture numérique, de pensée informatique ou encore de pensée computationnelle (Amadiou & Tricot, 2020).

En effet, la pensée informatique est l'un de ces concepts en vogue qui a gagné en intérêt au cours de ces dernières années. Pour certains, celle-ci devrait devenir un aspect essentiel dès l'école primaire, pour d'autres aucune valeur ajoutée n'y serait associée et les derniers n'en n'ont même jamais entendu parler. Alors, qu'entendons-nous par pensée informatique et pourquoi est-il judicieux d'englober ces compétences dans les programmes de l'enseignement primaire (El Niño, 2018) ?

Au cours des dernières années, l'intégration de l'informatique, que ce soit au niveau primaire ou secondaire, a été mise en évidence, notamment, suite à la nécessité d'obtenir une future main d'œuvre maîtrisant les compétences dans ce domaine (Chen et al., 2017). Bien qu'essentielles aux professionnels en informatique, les compétences en pensée informatique le sont tout autant en termes de compétences de la vie quotidienne (Wing, 2006). Considérées comme faisant partie intégrante de chaque aspect du travail ou de la vie, elles font désormais partie de nombreux curriculums à travers le monde, et ce, dès la maternelle (Wei et al., 2021).

Depuis les cinq dernières années, la France et la Suisse ont mis l'accent sur ces enseignements à tous niveaux en les intégrant officiellement dans leurs programmes scolaires (Broisin et al., 2021). Plus récemment, en 2020, le Luxembourg a, à son tour, ajouté la pensée informatique comme sujet d'apprentissage au niveau de son enseignement fondamental (Hennico et al, 2021). Si nous nous référons au rapport Eurydice, publié en 2019, évaluant les politiques éducatives en matière de numérique, nous pouvons classer la Belgique francophone dans le camp des « mauvais élèves » en termes de développement de ses compétences numériques tant au niveau primaire que secondaire. La réforme du Pacte pour un Enseignement d'Excellence devrait néanmoins pouvoir assouvir ces manquements (Henry et al., 2021).

L'un des objectifs liés à l'enseignement de la pensée informatique réside dans le fait de développer les capacités des apprenants à pouvoir utiliser les outils informatiques afin de résoudre des problèmes et de les préparer davantage à vivre dans une société qui se veut de plus en plus numérique (Yadav et al., 2016). Pour ce faire, les apprenants se doivent d'acquérir, de développer mais également de conserver un niveau de compétence en pensée informatique. Afin de permettre aux différentes compétences acquises de perdurer dans le temps, il est nécessaire d'explorer le champ des pratiques pédagogiques (Casanova et al., 2020), sans quoi le fait d'introduire des activités ciblant le développement de la pensée informatique auprès des jeunes élèves pourrait entraîner certains manquements dans l'acquisition des compétences (Tikva & Tambouris, 2021).

En outre, l'une des difficultés majeures à laquelle les enseignants se retrouvent confrontés dans ce domaine est le faible degré d'intérêt ou encore de motivation des filles pour le secteur de l'informatique et des technologies. Cette tendance présente de multiples origines. Tant les recherches de Buitrago Flórez et al., 2017 que celles de Kelleher et al., 2017 pointent du doigt le manque de soutien parmi les pairs comme possible cause de ce problème, tandis que Doubé & Lang (2012) visent le manque de motivation personnelle et/ou les stéréotypes de genre. Angeli et Valadines (2020), quant à eux, estiment qu'il existe un manque de contenu qui soit adéquat en termes de besoins et prenne en considération les intérêts des enfants. Ainsi, il convient de continuer à explorer ce champ de recherche afin de mieux comprendre comment aider les apprenants à assimiler au mieux ces compétences (Wei et al, 2021)

Toutefois, certains paramètres restent confus : l'âge adéquat pour l'apprentissage de chaque compétence, les concepts atteignables en fonction de l'année scolaire, les stratégies d'enseignement à adopter ainsi que l'efficacité de celles-ci sur les apprentissages sans oublier les stratégies d'évaluation pour assurer leur développement (Zapata-Caceres et al, 2021).

Ceci nous amène donc à définir notre question de recherche comme suit :

« L'implémentation de scénarios pédagogiques couplée à deux techniques d'étayage différentes peut-elle faciliter l'acquisition de compétences spécifiques liées à la pensée informatique et algorithmique chez des filles et des garçons du premier cycle de l'enseignement fondamental de l'enseignement ordinaire en Fédération Wallonie-Bruxelles ? »

Pour tenter de répondre à cette question, nous débuterons par l'éclairage de différentes notions relatives à la pensée informatique et algorithmique (PIA). Nous poursuivrons avec une synthèse et une analyse des différentes informations que nous retrouvons dans la littérature scientifique concernant les études menées en PIA et notamment le projet PIAF, les différences de genre dans les activités de programmation, l'évaluation de la PIA, ainsi que les enjeux et les défis relatifs à ce domaine.

Ensuite, nous créerons un scénario pédagogique que nous implémenterons dans une classe du premier cycle de l'enseignement primaire en proposant aux élèves deux techniques d'étayage différentes.

Par la suite, nous analyserons l'impact de cette scénarisation pédagogique et de ces deux procédures d'étayage sur le développement des compétences en PIA par l'intermédiaire de plusieurs outils d'évaluation: un pré-test, un post-test immédiat et un post-test différé permettant également d'évaluer le maintien de ces compétences dans le temps; une grille d'observation ainsi qu'un journal de bord.

Suite à l'implémentation de notre scénario, nous analysons les résultats obtenus à l'aide de la mise en place d'une méthodologie mixte et débattons des différentes hypothèses formulées. La discussion met en évidence tant les éléments de réponse relatifs à notre question de recherche que les perspectives à développer.

Pour conclure cette recherche, nous en retracerons le fil rouge et veillerons à alimenter modestement le débat par nos découvertes.

2. Revue de la littérature

2.1. Qu'est-ce que la pensée informatique ?

Souvent amalgamés, les différents termes « pensée informatique », « programmation informatique » ou encore « informatique » peuvent prêter à confusion ou encore à controverse (Grover & Pea, 2013). En effet, la littérature scientifique recense une multitude de définitions de la pensée informatique ainsi que plusieurs propositions sur les compétences et les aptitudes qui lui sont liées (Zhang et al., 2021).

La pensée informatique ne date pas d'hier, bien loin de là même puisque ce concept remonte aux travaux de Seymour Papert qui, en 1980, plaidait déjà au développement d'une pensée algorithmique par l'intermédiaire du langage de programmation LOGO (Papert, 1980). En 2006, Jeannette Wing donne un nouvel élan à la pensée informatique qu'elle appelle «computational thinking». Elle la considère comme une compétence fondamentale pour tout un chacun et pas uniquement pour les informaticiens. Au-delà de la lecture, de l'écriture ou encore de l'arithmétique, nous devrions ajouter la pensée informatique à la capacité d'analyse de chaque enfant. Elle spécifie que :

« La pensée informatique consiste à résoudre des problèmes, à concevoir des systèmes et à comprendre le comportement humain, en s'appuyant sur les concepts fondamentaux de l'informatique. La pensée informatique comprend une gamme d'outils mentaux qui reflètent l'étendue du domaine de l'informatique. » (Wing, 2006, p.33)¹

Selon Tchounikine (2017), les concepts de pensée informatique peuvent être scindés en cinq grandes catégories :

- appréhender un problème et sa solution à différents niveaux (abstraction) ;
- réfléchir aux tâches à accomplir sous forme d'une série d'étapes (algorithmes) ;

¹ “Computational thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. Computational thinking includes a range of mental tools that reflect the breadth of the field of computer science”.

- comprendre que pour résoudre un problème complexe il faut le décomposer en plusieurs problèmes simples (décomposition) ;
- comprendre qu'il est probable qu'un nouveau problème soit lié à d'autres problèmes déjà résolus par l'élève (reconnaissance de formes), et
- réaliser que la solution à un problème peut servir à résoudre tout un éventail de problèmes semblables (généralisation). (p.11).

Shute et ses collègues (2017) définissent quant à eux la pensée informatique comme "la base conceptuelle nécessaire pour résoudre des problèmes de manière efficace et efficiente, avec des solutions réutilisables dans différents contextes" ²(p.142). Ils classent les compétences en pensée informatique dans les quatre catégories que l'on retrouve le plus souvent dans la littérature scientifique, à savoir, abstraction, décomposition, algorithmes et débogage auxquelles ils ajoutent l'itération et la généralisation.

Force est de constater qu'à ce jour, nous ne pouvons acquiescer sur une définition consensuelle de la pensée informatique (Selby & Woollard, 2013). Ceci compliquant l'évaluation de ses compétences et la difficulté de conclure sur l'efficacité de tout outil pédagogique présumant de son développement (Sigayret et al., 2021).

² "The conceptual foundation required to solve problems effectively and efficiently (i.e., algorithmically, with or without the assistance of computers) with solutions that are reusable in different contexts".

2.2. Quelles compétences sont développées en pensée informatique et algorithmique ?

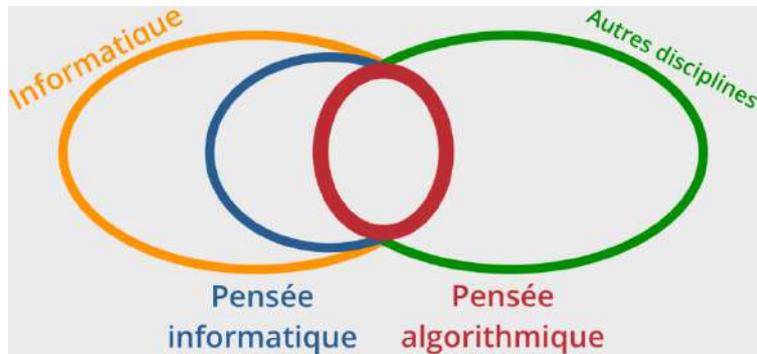


Figure 1 / Interactions en jeu (Denis, 2020, p.7)

Comme nous venons de le voir, les contours du concept de pensée informatique semblent compliqués à appréhender. La figure ci-dessus nous montre la complexité des liens qui existent entre l'informatique, la pensée informatique, la pensée algorithmique et les autres disciplines. C'est pourquoi certains pays se sont penchés sur l'élaboration de référentiels permettant ainsi de déterminer les compétences qui lui sont associées.

Basés sur la recherche de Parmentier et al. (2020), nous allons à présent faire un tour d'horizon de l'existant.

2.2.1. Les pays anglo-saxons

Il y a une quinzaine d'années, Les pays anglo-saxons se rendent compte de la nécessité de définir un cadre de formation qui se rapproche de la pensée informatique. Parmi eux, nous retrouvons les États-Unis avec le « *CSTA K12 standards*³ ». Rédigé en 2011, ce dernier a fait l'objet d'une révision en 2017 et détaille une série de compétences mobilisables dès l'âge de cinq ans et rassemblées autour de cinq axes :

1. Systèmes informatiques
2. Réseaux et internet
3. Données et analyse
4. Algorithmes et programmation
5. Impact de l'informatique

³ Voir <https://www.csteachers.org/page/about-csta-s-k-12-nbsp-standards>

Bien que ce curriculum soit relativement orienté d'un point de vue technique, l'approche CSTA apporte néanmoins une contribution substantielle puisque son référentiel se veut fondé scientifiquement, validé par des personnes présentes sur le terrain, assorti d'une évolution dans l'acquisition des compétences en fonction de l'âge des apprenants tout en s'insérant dans un processus élargi aux niveaux d'études supérieurs.

Ensuite, nous retrouvons l'Angleterre avec le « *Computing Progression Pathways* ⁴ » qui, lui, repose sur six domaines :

1. Algorithmes
2. Programmation
3. Données et représentations
4. Matériel et traitements
5. Communication et réseaux
6. Technologies de l'information

Ce référentiel a pour objectif d'être détaillé, ce qui fait sa richesse d'une part et sa principale limite d'autre part puisque bon nombre des compétences qui le composent se rattachent à plusieurs domaines qui peuvent ainsi engendrer un manque de compréhension.

2.2.2. L'australie

C'est en 2015 que l'Australie développe son « *F-10 curriculum* ⁵ » autour de deux disciplines inculquées aux enfants dès l'âge de six ans : les technologies de l'information et de la communication, d'une part, et les technologies numériques, d'autre part. Ces dernières font référence aux compétences spécifiques à l'informatique, en ce compris la pensée informatique, et se développent autour de dix axes :

- | | |
|-------------------------------|------------------|
| 1. Abstraction | 6. Spécification |
| 2. Systèmes numériques | 7. Algorithmes |
| 3. Représentation des données | 8. Implantation |
| 4. Collection des données | 9. Impact |
| 5. Interprétation des données | 10. Interactions |

⁴ Voir <https://community.computingatschool.org.uk/resources/2324/single>

⁵ Voir <https://www.australiancurriculum.edu.au/f-10-curriculum/>

Bien qu'il permette à chaque apprenant de développer sa pensée informatique de manière précoce, une vision trop technique de l'informatique pourrait avoir des conséquences sur son abordabilité ou encore véhiculer des stéréotypes de genre face à ce domaine.

2.2.3. L'Europe

Par ailleurs, en 2013, l'Europe se lance dans la rédaction de son référentiel de compétences numériques destiné à tout citoyen et publie pour la première fois son « *DigComp Citizen*⁶ » qui sera mis à jour par la suite en 2017. Elle édite également « *DigCompEdu* » destiné aux compétences numériques des enseignants.

DigComp 2.1 repose sur cinq domaines de compétences (Carretero, 2017) :

1. Littératie des données et de l'information
2. Communication et collaboration
3. Création de contenus numériques
4. Sécurité
5. Résolution de problèmes

Chaque domaine est lui-même divisé. On retrouve au total 21 sous-domaines pour lesquels huit niveaux de maîtrise (basés sur des savoirs, savoir-faire et savoir-être) ont été définis.

Ce référentiel présente également plusieurs atouts relatifs au développement de la pensée informatique. En effet, par le biais d'exemples concrets, il fait le lien avec les situations de la vie quotidienne et permet d'impacter les politiques éducatives européennes de par son rôle normatif. Néanmoins, nous pouvons relever un manque de précision en termes de compétences liées à la pensée informatique qui, de plus, se retrouvent perdues dans la masse.

Outre le fait de s'être rendu compte de l'importance de développer les compétences numériques dès le plus jeune âge et d'y apporter notamment des progressions en termes de maîtrise et de formation, les référentiels que nous venons d'aborder présentent également certaines limites.

En effet, hormis dans le cadre australien, nous ne pouvons discerner clairement les compétences de la pensée informatique pouvant être considérées comme des compétences transférables à d'autres disciplines. De plus, du fait de la richesse et la complexité des progressions proposées, ces référentiels ne font pas nécessairement sens pour des non-spécialistes (Parmentier et al., 2020).

⁶ Voir <file:///C:/Users/pasca/Downloads/KE-02-15-657-EN-N.pdf>

Pour faire suite à ce constat, différents chercheurs européens (Parmentier et al., 2020) ont, dans le cadre d'un projet Erasmus+, pris appui sur l'existant pour créer de manière collaborative un nouveau référentiel poursuivant trois objectifs. Premièrement, qu'il soit centré précisément sur la pensée informatique. Deuxièmement, qu'il soit ajusté à l'enseignement fondamental tout en asseyant des fondements qui soient en accord avec les niveaux secondaire ou encore supérieur. Enfin, qu'il soit abordable pour des personnes non expertes.

Le référentiel de compétences élaboré dans le cadre du projet Erasmus+ PIAF (cf. annexe 1, p. 105) fait état de six compétences principales et de vingt-huit sous-compétences elles-mêmes réparties en deux niveaux : les sous-compétences de base qui visent les enfants âgés de 5 à 8 ans et les sous-compétences avancées qui visent les enfants âgés de 9 à 12 ans. Les six compétences principales de la pensée informatique et algorithmique sont les suivantes :

- C 1 Définir des abstractions / généraliser
- C 2 Composer/décomposer une séquence d'actions
- C 3 Contrôler une séquence d'actions
- C 4 Évaluer des objets ou des séquences d'actions
- C 5 Manipuler des représentations formelles
- C 6 Construire une séquence d'actions de manière itérative

Pour accompagner ce référentiel, l'équipe de recherche en charge de son élaboration a également créé un glossaire permettant d'éclaircir les termes plus techniques ainsi qu'un répertoire d'exemples concrets permettant aux enseignants notamment de se familiariser avec les différentes sous-compétences.

2.3. Quelles approches sont utilisées pour développer les compétences en pensée informatique ?

Tout en continuant rapidement son développement, l'enseignement de la pensée informatique présente en outre des dispositifs de mise en œuvre avancés et innovants (Heintz et al., 2016, as cited in Menon, 2020). Nous pouvons mettre en avant trois approches utilisées pour permettre aux élèves d'aborder la pensée informatique au niveau scolaire : l'informatique débranchée, les outils de programmation visuelle ainsi que les activités de robotique pédagogique dites "branchées" (Menon, 2020).

Nous allons à présent nous pencher sur le contenu de ces activités.

2.3.1. L'informatique débranchée

Les activités débranchées ne requièrent aucun outil numérique.

Mais l'ordinateur est-il indispensable pour éduquer à la pensée informatique ?

Romero et al. (2018) font état de différentes plus-values des activités débranchées. Premièrement, tel que Sweller (1994) l'a noté, de par sa technicité, l'utilisation d'une machine augmente la charge cognitive qui limite la mémoire de travail alors incapable de traiter l'information nouvelle. Les activités débranchées sont par ailleurs moins impressionnantes tant pour les élèves que pour les enseignants étant donné leur pratique dans d'autres domaines. Deuxièmement, le fait d'utiliser son corps ou de manipuler des objets du quotidien en fait une activité engageante autant au niveau physique que cognitif puisque les différents types de mémoires procédurales, épisodiques et sémantiques sont en interaction. Enfin, la construction d'une analogie tangible des notions conceptuelles en informatique est placée au centre de leur intérêt.

Faber et ses collègues (2017) ont quant à eux menés des recherches sur le design du dispositif didactique mis en place dans les activités débranchées. Pour se faire, quinze enseignants ont dispensé un cours visant le développement de la pensée informatique au travers d'activités débranchées auprès de 411 élèves scolarisés dans 26 écoles implantées aux Pays-Bas.

Ils ont été soumis à un focus group à la suite duquel les chercheurs ont pu formuler une série de préconisations, notamment, le fait de proposer aux élèves différentes activités qui puissent répondre à leur niveau de compétence ou encore l'explicitation claire de la manière dont les différents concepts seront utilisés par la suite dans les activités dites "branchées".

En 2020, del Olmo-Muñoz et al. ont mené une étude quasi-expérimentale destinée à évaluer l'impact de l'inclusion d'activités débranchées sur le développement des compétences en pensée informatique, sur la motivation ainsi que l'influence du genre dans ces deux domaines. L'implémentation s'est déroulée en deux phases auprès de 84 élèves scolarisés en deuxième année de l'enseignement primaire espagnol : lors de la première phase, un groupe travaillait avec des activités débranchées, tandis que l'autre groupe avec des activités branchées. Lors de la deuxième phase, tous deux étaient confrontés à des activités branchées. Ils arrivent à conclure que le fait de proposer aux élèves un enseignement basé sur des activités débranchées ou mixtes est non seulement bénéfique en termes d'acquisition des compétences en pensée informatique mais également en termes de motivation.

2.3.2. Les activités de programmation visuelle

L'utilisation d'outils de programmation a notamment comme objectif le développement de compétences informatiques créatives. Depuis leur introduction dans ce domaine, nous avons pu constater l'apparition de différentes applications de codage ou langages de programmation tels que *Scratch*, *Hopscotch* ou encore *Kodable* (Menon, 2020).

Parmi ceux-ci, les études concernent particulièrement Scratch (Resnick & Siegel, 2015) et la manière dont il est utilisé.

La rétroaction instantanée fournie par l'outil numérique ainsi que la multitude de potentialités qui lui sont conférées seraient enclines à permettre aux élèves de développer leurs compétences en PIA. Néanmoins, tel que nous l'avons souligné plus haut, la charge cognitive associée à l'utilisation des logiciels pourrait affecter l'apprentissage en diminuant son attention (Sigayret et al., 2021 ; Sweller, 1994).

2.3.3. Les activités de robotique pédagogique dites « branchées »

La pensée informatique en rapport avec la robotique fait partie des champs de recherche relativement récents (Stewart et al., 2021). Les études actuelles qui ont investigué ce domaine nous indiquent que les activités de robotique pédagogique constituent une approche intéressante pour développer la pensée informatique des jeunes enfants. En effet, ils peuvent interagir directement avec un robot et rendre immédiatement compte du résultat de leurs interactions sur le comportement de celui-ci. De plus, ces activités permettent de rendre certaines notions plus concrètes que si elles étaient enseignées de manière théorique ou traditionnelle (Beraza et al., 2010 ; Bers et al., 2014, Stoeckelmayr et al., 2011).

Roméro, Noirpoudre et Viéville (2018) ont synthétisé une cinquantaine d'études internationales dans le domaine des sciences de l'éducation et se sont intéressés, notamment, à l'utilité de la robotique pédagogique. Ils approuvent d'abord la faisabilité de l'immiscer dans le contexte scolaire et ce même auprès des plus jeunes. Ils ajoutent cependant que pour juger de l'impact positif de la robotique pédagogique sur les apprentissages, il ne suffit pas de viser les savoir-faire technologiques mais bien de dépasser ce niveau afin de pouvoir transférer les compétences acquises dans d'autres apprentissages, et ce, même chez les élèves présentant des difficultés.

Le lien entre la robotique pédagogique et la pensée informatique a été, entre autres, identifié par Atmatzidou et Demetriadis (2016) qui observent que le développement de ces compétences nécessite du temps et que les élèves atteignent le même niveau de performance quel que soit leur âge ou leur genre avec cependant le fait qu'il faille accorder plus de temps aux filles

comparativement aux garçons. Pour ce faire, ils ont proposé à 164 étudiants âgés de quinze et dix-huit ans des activités de robotique pédagogique à raison de deux heures par semaine et ce, pendant onze semaines. Répartis par groupes de trois ou quatre étudiants, ils étaient amenés à réaliser des tâches de programmation de complexité croissante guidées par des fiches de travail. Ils ont ensuite été évalués à différents moments de l'étude grâce à l'utilisation d'outils faisant usage de modalités écrites et orales en mettant l'accent sur les concepts d'abstraction, de généralisation, d'algorithme, de modularité et de décomposition.

En 2014, Bers, Flannery et leurs collègues s'intéressent aux plus petits et mènent leur étude auprès de 63 enfants âgés en moyenne de cinq ans et demi. Par l'intermédiaire de leur programme « *TangibleK Robotics* », les apprenants sont amenés à développer leurs compétences en pensée informatique grâce à des activités mêlant robotique, programmation et résolution de problèmes selon une approche constructiviste. Dans son ensemble, cette étude montre l'intérêt que les élèves de maternelle peuvent témoigner à l'égard de ce genre d'activités et leur capacité à pouvoir se saisir des notions de robotique qui leurs sont appropriées. L'implémentation de ce programme démontre que les jeunes enfants peuvent faire leurs premiers pas dans le développement de leurs compétences en pensée informatique à partir du moment où les outils mis à leur disposition, le curriculum et les pédagogies proposées sont en adéquation avec l'âge des apprenants.

Plus récemment encore, pour donner suite à leur revue de littérature systématique en lien avec l'utilisation de la robotique éducative au service d'autres apprentissages, Spalaôr et Benetti (2017) aboutissent à plusieurs reprises à la conclusion que celle-ci représente un réel intérêt surtout lorsque cette approche est couplée à une perspective constructiviste (Spalaôr & Benetti, 2017, as cited in Romero et al., 2018).

Dans ce paradigme, les apprenants sont acteurs de la construction de leur savoir et « *ils sont épistémologues c'est-à-dire qu'ils ont la capacité d'effectuer une étude critique de leur propre réflexion* » (Strebelle et al., 2017, p. 207).

En outre, Baek et al. (2019) font état d'une acquisition significative des compétences en pensée informatique des apprenants lorsque ceux-ci sont engagés dans des activités de codage via la robotique pédagogique.

Enfin, n'oublions pas de préciser, bien que certains robots constituent un outil facile d'utilisation avec de jeunes enfants, il est cependant nécessaire pour les enseignants d'apprendre

à l'utiliser de manière appropriée afin de maximiser ses effets sur le développement de la pensée informatique de ceux-ci (Angeli & Valanides, 2020).

2.4. Quelles sont les stratégies de programmation mises en œuvre par les apprenants ?

Dans leur étude de cas, Komis et Misirli (2013) avaient notamment comme objectif d'étudier les stratégies de programmation mises en œuvre par 108 enfants de maternelle âgés entre quatre et six ans. Ils ont utilisé un scénario pédagogique conçu dans le cadre d'un projet européen et basé sur le robot pédagogique Bee-Bot. Les apprenants sont amenés à construire une séquence, à exécuter des trajets dans l'espace et à utiliser les commandes de gestion telles qu'effacer et démarrer au travers de la programmation et avec l'aide potentielle de cartes de programmation.

Contrairement aux travaux de Greff (1998) et de Demo et al. (2008) prenant comme système de référence le corps de l'enfant, les enfants apprennent ici « *à contrôler, à manipuler et à programmer le Bee-Bot de manière organisée et construisent les concepts de direction et d'orientation (Avancer, Reculer, Tourner à droite, Tourner à gauche) durant l'estimation, la comparaison et la construction des trajets précis [...] Il s'agit, dans ce cas, d'une fonction cognitive de haut niveau* » (Komis & Misirili, 2013, p.2).

Le scénario pédagogique prévoit deux stratégies possibles (Komis & Misirili, 2013, p.5):

- La stratégie totale dans laquelle l'apprenant compose, programme, et, ensuite, teste l'ensemble de l'algorithme qu'il a créé et ce "*selon la règle CLEAR – ensemble des commandes d'orientation et de direction (AVANCER, TOURNER, RECULER) – GO*". Il s'agit de la stratégie la plus rencontrée.
- La stratégie pas-à-pas dans laquelle l'apprenant entre une seule commande à la fois, et, ensuite, la teste immédiatement et ce "*selon la règle CLEAR - une commande de direction ou d'orientation – GO, CLEAR - une commande de direction ou d'orientation - GO*".

Outre ces deux stratégies, les chercheurs en ont finalement recensées cinq autres (Komis & Misirili, 2013, p.5) :

- La stratégie de sous-programme dans laquelle l'apprenant va fractionner l'ensemble de son programme en sous-programmes comprenant à chaque fois un programme complet de programmation, c'est-à-dire "*un ensemble de commandes de direction et*

d'orientation entre la commande CLEAR et la commande GO". Il s'agit de la stratégie la plus intéressante selon les chercheurs.

- La stratégie totale et pas à pas dans laquelle l'apprenant *"commence par la stratégie Totale et il termine par la stratégie Pas à pas"*.
- La stratégie totale et sous-programme dans laquelle l'apprenant *"commence par la stratégie Totale et il termine par la stratégie Sous-Programme"*.
- La stratégie sous-programme et pas-à-pas dans laquelle l'apprenant *"commence par la stratégie Sous-Programme et il termine par la stratégie Pas à pas"*.
- La stratégie par essais et erreurs dans laquelle l'apprenant *"n'a pas verbalisé l'algorithme et il n'a pas rédigé son programme par cartes de commandes. Il ne dispose donc pas une représentation mentale de l'algorithme et par conséquent il approche par essai et erreur pour trouver une solution"*.

2.5. Comment évaluer les compétences en pensée informatique et algorithmique ?

Endéans les vingt dernières années, de multiples instruments destinés à mesurer le niveau d'acquisition des compétences en pensée informatique ont vu le jour. Néanmoins, la plupart des outils d'évaluations conçus jusque maintenant s'adressent généralement à des enfants plus âgés ou encore à des adultes (Román-González et al., 2018 ; Werner et al., 2012 ; Chen et al., 2017).

En effet, peu d'entre eux ont trait aux plus jeunes fréquentant le début de l'école primaire, âgés pour la plupart de quatre à neuf ans. La majorité des travaux préalables se concentrent sur l'utilisation de protocoles d'entretien ou d'évaluations de codage basées sur des projets. Bien qu'elles permettent de mettre en lumière la pensée des enfants, leur format et leur caractère chronophage les rendent difficilement réalisables en dehors d'un cadre de recherche.

De plus, la verbalisation des processus mentaux est difficile chez les jeunes enfants, ce qui peut se poser également comme limite à ce genre type d'instrument de récolte de données (Relkin et al., 2020).

En outre, que ce soit pour les chercheurs, les enseignants ou encore les apprenants, l'évaluation des compétences en pensée informatique permet de livrer des informations qui peuvent s'avérer précieuses quant à l'efficacité de programmes éducatifs spécifiques ou encore d'interventions plus ponctuelles (Resnick, 2007 ; Sullivan and Bers, 2016).

Afin de combler ce manque en la matière, Relkin et al. (2020) ont mis au point “TechCheck”, un nouvel instrument validé auprès de 768 élèves âgés de 5 à 9 ans ayant participé à une recherche impliquant des activités de codage et permettant l'évaluation des compétences en pensée informatique chez les jeunes enfants : algorithmes, modularité, structures de contrôle, représentation, matériel/logiciel et débogage. Celui-ci présente plusieurs avantages : il ne requiert notamment aucune connaissance préliminaire en termes de programmation informatique, se veut débranché, est construit sur un format de questions à choix multiples et permet d'être administré simultanément à un grand nombre d'apprenants, en ligne ou en personne, et ce, en moins de quinze minutes.

Globalement, “TechCheck” présente de bonnes propriétés psychométriques, est simple à implémenter et à évaluer, et permet de distinguer les élèves en fonction de leurs compétences en pensée informatique (Relkin et al., 2020).

Sigayret et al. (2021), ont concentré, quant à eux, leur attention sur les outils permettant une évaluation quantitative. Ils les classifient en trois catégories :

- Le Computational Thinking Scales (CTS) proposé par Korkmaz et al. (2017) regroupe des items provenant de différents tests déjà existants et synthétisés autour de cinq facteurs : créativité, pensée algorithmique, coopération, esprit critique et résolution de problème. Statistiquement parlant, cet instrument a été validé et permettrait de distinguer les niveaux de pensée informatique des élèves. Par ailleurs, il convient de noter que cette étude a été menée auprès d'un public universitaire.
- L'utilisation de certains blocs émanant des outils de programmation visuelle pourrait, selon certains, être corrélée à l'intégration des différents concepts relatifs à la pensée informatique (Moreno-Léon & Robles, 2015 ; da Cruz Alves et al., 2019).
- Enfin, l'élaboration de tâches de résolution de problèmes permettrait également de pouvoir juger de la maîtrise des compétences en PIA. Citons notamment le Computational Thinking Test (CTt) de González (2015) destiné aux élèves âgés entre dix et seize ans.

Toujours selon ces mêmes auteurs, ces outils ne permettent toutefois pas de pouvoir différencier de façon distincte les connaissances des élèves d'une part, et leur aptitude à transférer celles-ci pour solutionner des problèmes plus difficiles d'autre part. Cette différenciation est, pour eux, essentielle et c'est pourquoi ils proposent, sur base de ces observations, deux nouveaux tests : un test pour mesurer la compréhension des notions de base en programmation construit sur un format de questions à choix multiples et un test pour mesurer leur aptitude à résoudre des

problèmes algorithmiques construit sur un format de résolution de problèmes. Ceux-ci ont été testés auprès d'élèves francophones scolarisés de la troisième à la sixième année de l'enseignement primaire. Bien que présentant des résultats prometteurs, leur fiabilité et leur validité n'a pas pu être prouvée.

Revenons néanmoins sur le CTt développé par González (2015). Celui-ci se distingue comme étant un instrument de mesure autonome, affranchi d'un quelconque outil de programmation et conçu selon approche psychométrique fiable et valide. Cette base leur a permis de se tourner vers les plus jeunes en développant le Beginners Computational Thinking Test (BCTt). Une première version du test (BCTt v.1) a été conçue et adaptée tant sur sa forme que sur son contenu, puis ensuite soumise à un processus de validation par des experts (Zapata-Cáceres et al., 2020). Au nombre de quarante-cinq, ces derniers ont validé l'instrument en jugeant notamment le niveau de difficulté, la pertinence de chaque item au regard de la mesure des compétences en pensée informatique, la durée du test ou encore la qualité de l'interface.

Après avoir tenu compte des résultats, des conseils ainsi que des obstacles rencontrés, le test a été revu dans une seconde version améliorée (BCTt v.2).

Ce dernier se compose de 25 items construit sur un format de questions à choix multiples organisées par ordre de difficulté croissante. Afin de diminuer la probabilité de réponse au hasard, le nombre de propositions est porté à quatre.

Les différents items mesurent les concepts de séquençage (6 items), de boucle simple (5 items), de boucle imbriquée (7 items) et de condition (7 items).

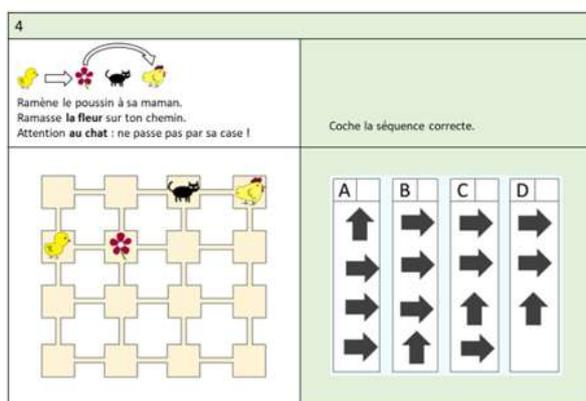


Figure 2 / Exemple de question

2.6. Quels sont les enjeux et les défis relatifs à la pensée informatique ?

Alors qu'il est abondamment accepté dans la littérature scientifique que la pensée informatique implique toute une série de compétences telle que la décomposition de problème, le développement d'algorithmes et l'abstraction, il n'existe encore que peu de preuves en ce qui concerne les différents enjeux et défis à prendre en considération lors de l'élaboration d'expériences d'apprentissages en pensée informatique (Angeli & Giannakos, 2020). Ces derniers auteurs ont présenté plusieurs études en lien avec les différents aspects de la recherche en pensée informatique tout en faisant état des défis à relever dans le domaine. Celles-ci, appuyées également par d'autres auteurs, nous permettent notamment de fournir plusieurs indications concernant : a) la mise à l'épreuve d'activités de pensée informatique qui puisse permettre d'accroître l'autonomie des filles et des garçons, b) le développement des compétences en pensée informatique des jeunes enfants par l'usage de techniques d'étayages et de la robotique pédagogique, c) l'importance d'avoir recours à l'expérimentation empirique dans le but de faire avancer la recherche dans le domaine de l'enseignement de la pensée informatique.

2.6.1. Les stratégies d'apprentissage pour encourager le développement de la pensée informatique en fonction du genre

Plusieurs recherches (Bonomo, 2010 ; King & Gurian, 2006 ; Gurian et al., 2011) ont été effectuées dans le domaine des neurosciences et ont révélé le fait que l'hippocampe, siège des fonctions mnésiques et langagières, se développe plus vite et est plus volumineux chez les filles par rapport aux garçons. Ceci aurait comme conséquence un impact sur les compétences de séquençage, de vocabulaire, de lecture et d'écriture. Les filles bénéficieraient d'une année et demie d'avance sur les garçons en matière de lecture et d'écriture.

Les différences au niveau du cortex cérébral se marquent davantage chez les garçons puisqu'une plus grande partie de celui-ci est défini pour les relations spatiales. Par conséquent, ils retireraient un meilleur apprentissage par le mouvement et l'expérience visuelle. Sur base de ces différences biologiques, les filles seraient plus enclines à se tourner vers des activités de coopération dans lesquelles elles peuvent travailler en groupe et partager leurs opinions tandis que les garçons préféreraient les activités individuelles, qui vont vite, dont les informations passent par le corps et qui sont centrées sur l'espace et la manipulation (Gurian, 2003, as cited in Angeli & Valanides, 2020). De plus, les filles ont davantage d'aires corticales dédiées aux

fonctions verbales, ont un registre sensoriel mieux développé, sont plus aptes à rester assises et à écouter (Bonomo, 2010).

Les recherches menées par Gurian et al. (2011) et Stevens-Smith & Fleming (2016) font état d'une différence au niveau de l'hémisphère dominant chez les filles et chez les garçons. En effet, l'hémisphère gauche est associé davantage aux aptitudes verbales, à l'expression orale, à la lecture, à l'écriture, ainsi qu'aux mathématiques ; tandis que l'hémisphère droit est associé davantage aux aptitudes spatiales, à la manipulation d'objets, aux symboles, aux images ainsi qu'à la prise de risque. La plupart des garçons auraient une disposition à avoir une dominance du cerveau droit, alors que les filles auraient deux hémisphères plus équilibrés leur permettant de faire preuve d'une meilleure organisation et d'une plus grande focalisation sur les détails.

Ces différences se marquent également au niveau des matières scolaires privilégiées et dans lesquelles ils obtiennent de meilleurs résultats. Puisqu'une plus grande partie de leur cerveau leur est consacrée, les garçons se distinguent dans les matières scientifiques et en mathématiques (Benbow & Lubinski, 2007). Les filles, quant à elles, excellent en littérature (lecture, écriture et langage) car la zone responsable de ces fonctions (l'hippocampe) se développe plus tôt et plus largement (Gurian et al., 2011).

Kommer (2006) incite les enseignants à prendre ces différences en considération et à pouvoir répondre de manière ciblée aux besoins particuliers de chaque genre. Dans cette optique les garçons bénéficieraient de plus d'espace et de possibilités de pouvoir bouger tandis que les filles pourraient collaborer et s'exprimer davantage. Il ajoute néanmoins que le fait de respecter les différences entre les deux genres ne doit pas entraîner une scission des classes ou encore la création de classes homogènes.

Blanchette Sarrasin & Masson (2017) classent quant à eux cette dominance hémisphérique dans le camp des neuromythes. Entendons par là, toute une série de fausses croyances sur le fonctionnement du cerveau. Ils évoquent le fait que certaines personnes soient plus compétentes dans un domaine ou dans un autre mais lorsqu'ils se posent la question de savoir si ces différences interindividuelles trouvent une explication dans la présence d'une dominance hémisphérique, ils répondent par la négative. Nielsen et al. (2013) ont mis à l'épreuve la validité de cette hypothèse auprès de 1011 sujets âgés entre 7 et 29 ans et concluent qu'il est impossible de procéder à une telle classification (cerveau gauche ou cerveau droit) en se basant sur le fonctionnement et la connectivité des hémisphères cérébraux.

Plusieurs études se sont intéressées à l'impact du genre sur le développement des compétences en pensée informatique. Certaines ont démontré qu'il n'existe pas de relation significative entre les deux (Werner et al., 2012, as cited in Durak & Saritepeci, 2018). Le genre est également amplement discuté dans le contexte de l'enseignement de la pensée informatique par le biais d'activités de programmation mais les résultats sont contradictoires (Sun et al., 2022).

En 2016, Atmatzidou et Demetriadis (dont nous avons déjà évoqué les recherches dans le point 2.3.3.) s'intéressent au développement des compétences en pensée informatique en fonction de l'âge et du genre. La première observation qu'ils font est qu'ils ne constatent aucune différence liée à l'âge d'un point de vue du développement de leurs compétences (abstraction, généralisation, algorithmes, modularité et décomposition). Ils remarquent néanmoins que les filles semblent avoir besoin de plus de temps pour atteindre le même niveau de compétences que les garçons. Cette observation a également été faite dans l'étude menée par del Olmo-Muñoz et al. (2020) dont nous avons déjà parlé plus haut dans le chapitre 2.3.1.

Par ailleurs, à la suite de leur recherche menée par Román-González et al. (2017) auprès de 1251 élèves espagnols âgés entre dix et seize ans et dont l'objectif était de tester un nouvel instrument d'évaluation de la pensée informatique (le Computational Thinking Test), les chercheurs font état d'un meilleur développement des compétences en faveur des garçons avec néanmoins un écart progressif entre les genres au fur et à mesure que les apprenants avancent dans leur cursus scolaire.

Pour leur part, Papavlasopoulou, Sharma et Giannakos (2020) ont élaboré et testé un atelier destiné aux élèves de la maternelle à la fin de l'enseignement primaire afin de leur apprendre à coder. La conception et le développement d'activités susceptibles d'étayer les différents concepts liés à la pensée informatique et la motivation aussi bien des filles que des garçons s'avèrent cruciaux pour l'enseignement et l'apprentissage des compétences relatives à la pensée informatique. Leur objectif était d'analyser les différences éventuelles en fonction du genre en utilisant une technique d'oculométrie couplée à toute une série de données qualitatives relevées lors d'entretiens avec les apprenants. Les résultats de l'oculométrie n'ont pas permis de mettre à jour une quelconque différence entre les filles et les garçons d'un point de vue des apprentissages au cours de l'activité de pensée informatique. Cependant, l'analyse des données qualitatives a permis de mettre en lumière certaines différences. Premièrement, au niveau des stratégies et des pratiques mises en place pendant l'activité de codage, et deuxièmement dans les perceptions de ces activités. « Preuve » est ici faite que les filles ne diffèrent pas par rapport aux garçons, que ce soit au niveau de leurs compétences ou de leur comportement, mais bien

qu'elles ont simplement une approche et des stratégies différentes lors des activités de pensée informatique. Il est donc important de prendre en compte ces notions lors de la création de ces activités afin de pouvoir aider au mieux les filles à maîtriser les différentes compétences relatives à la pensée informatique (Papavlasopoulou et al., 2020).

En conclusion, la question de potentielles différences de genre n'en reste pas moins importante à prendre en compte pour les enseignants dans la mise en place d'un environnement d'apprentissage et de techniques d'étayages qui seraient mieux appropriées aux filles ou aux garçons.

2.6.2. L'usage de techniques d'étayage

L'instruction par échafaudage, appelée aussi étayage, fait partie des stratégies d'enseignement qui permettent d'aborder le développement cognitif. Descendant tout droit de la théorie socioculturelle de Lev Vygotsky (1978) et plus particulièrement du concept de zone proximale de développement (ZPD), l'étayage s'appuie sur deux principes : premièrement, l'apprenant apprend au contact des personnes présentes dans son entourage et non de manière isolée. Deuxièmement, la ZPD se définit comme étant l'écart entre le niveau réel de ce que l'apprenant sait faire par lui-même et celui auquel il pourrait accéder avec l'aide d'autrui compétent.

Berger (2010) recense six conditions *sine qua non* pour qu'un étayage puisse être profitable à tout un chacun :

1. il faut éveiller l'intérêt de l'apprenant pour mener à bien l'apprentissage ;
2. si ce dernier s'avère complexe, il vaut mieux limiter le nombre d'étapes à réaliser en le facilitant ou en effectuant certaines d'entre elles nous-même ;
3. il faut guider l'enfant quant aux choix stratégiques à opérer ;
4. il faut maintenir le niveau de motivation tout au long du processus ;
5. il faut prévoir et informer l'apprenant des erreurs qu'il pourrait commettre afin de l'aider dans leur prévention ou leur correction éventuelle ;
6. il faut éveiller l'apprenant au désir de réussite en plaçant l'erreur au centre même du processus d'apprentissage.

Il ajoute qu'il est important pour les apprenants qui n'arriveraient pas à réaliser la tâche qui leur est demandée, de leur indiquer les solutions tout en décortiquant les différentes étapes nécessaires pour y arriver.

Notons qu'il existe plusieurs sortes d'étayage : l'étayage individuel, lors duquel un enseignant apporte son aide à l'apprenant en l'adaptant immédiatement aux besoins de celui-ci ; l'étayage

par les pairs lors duquel d'autres apprenants viennent fournir leur soutien ou lorsque celui-ci est fourni par un apprenant faisant preuve d'une certaine expertise envers un autre moins expérimenté ; la programmation par paire lors de laquelle les apprenants partagent l'outil de programmation tout en collaborant d'un point de vue cognitif pour réaliser la tâche qui leur est demandée ; l'étayage informatique (Carlborg et al., 2019).

L'étayage, tout comme la modélisation, le coaching ou encore le démantèlement font partie des méthodes qui permettent de contribuer au développement de l'expertise relatif à la théorie de l'apprentissage expérientiel (Carlborg et al., 2019).

2.6.2.1. L'apprentissage collaboratif

L'apprentissage collaboratif concerne des groupes d'apprenants travaillant en interaction dans le but de résoudre un problème, de réaliser une tâche demandée ou encore de créer un produit par l'intermédiaire de phases de négociation et de partage d'informations (Zhong et al., 2017).

Celui-ci fait partie des approches ayant pu montrer leur efficacité pour aider les apprenants à acquérir et à développer leurs compétences en pensée informatique (Hanks, 2008).

Parmi les diverses stratégies pédagogiques de l'apprentissage collaboratif, la programmation par paire s'avère être une stratégie qui a pu montrer son efficacité dans l'enseignement de la programmation, notamment chez les jeunes apprenants (Zhong et al., 2017).

Tantôt conducteur, tantôt navigateur, le rôle des partenaires s'échange régulièrement. Le conducteur est la personne qui va opérer les actions de programmation. C'est lui qui va agir sur l'outil de programmation ou encore écrire les différentes opérations. Tandis que le navigateur quant à lui va observer le travail de son binôme, lui fournir l'aide dont il aurait besoin et lui proposer des améliorations ou encore une réflexion stratégique sur le problème qui leur est posé (Wei et al., 2021).

Ainsi, la programmation par paire comporte plusieurs avantages. Tout d'abord, elle permet d'améliorer les compétences en programmation ainsi que la qualité des séquences établies et ce grâce à une compréhension plus approfondie due au partage d'informations et aux négociations qui en découlent (Dongo et al., 2016; Liebenberg, 2012). Ensuite, elle permet de réduire le sentiment de frustration et l'anxiété des apprenants débutants en programmation et engendre plaisir, sentiment de compétence et attitude positive face à l'apprentissage (Dongo et al., 2016). Enfin, elle permet d'accroître la motivation (Yang et al., 2016). Par ailleurs, notons que la programmation par paire permet également d'améliorer les compétences communicationnelles

(D'Angelo & Begel, 2017), collaboratives (Lewis, 2011) et relatives à la pensée informatique (Denner et al., 2014).

Cependant, il convient également de souligner les limites de l'application de la programmation par paire chez les jeunes apprenants ainsi que chez les débutants.

Chong et Hurlbutt (2007) ont remarqué une tendance au désengagement et un moindre apprentissage du partenaire présentant une faible expérience dans cette pratique collaborative tandis que, dans leur étude, Denner et ses collaborateurs (2019) ont observé que les apprenants ont été affectés lorsqu'ils se trouvaient en interaction avec des partenaires montrant moins d'intérêt ou étant moins enclins à collaborer.

Récemment, Wei et al. (2021) se sont intéressés à la manière dont ils pourraient mieux concevoir la programmation par paire pour ainsi augmenter les opportunités d'apprentissages des jeunes apprenants. Ils se sont interrogés sur l'efficacité de la programmation partielle par paire sur le développement des compétences en pensée informatique ainsi que sur le sentiment d'auto-efficacité des élèves du primaire. Ainsi, plutôt que d'insister sur le partage équitable des différents rôles, ils se sont plutôt attachés à l'aspect collaboratif de la programmation en duo en demandant aux apprenants de travailler en collaboration tout en proposant leurs productions de programmation individuelles.

Leur étude portait sur 171 élèves répartis dans quatre classes de quatrième primaire. Ceux-ci ont suivi un cours dispensé par le même enseignant utilisant le langage de programmation *Scratch*, et ce, pendant un semestre.

Les résultats de cette étude suggèrent que le recours à la programmation partielle par paire s'avère efficace pour améliorer tant les compétences en pensée informatique que le sentiment d'auto-efficacité des élèves de la maternelle à la fin de l'enseignement primaire.

2.6.2.2. L'usage de techniques d'étayages et de la robotique pédagogique.

Comme nous pouvons le constater dans la littérature existante, il est important d'administrer un étayage approprié aux enfants pendant leur apprentissage (Klein et al., 2000). Cet étayage devrait jouer un rôle majeur dans l'acquisition des compétences relatives à la pensée informatique.

En 2020, Angeli et Valadines décident de pousser leurs recherches dans le but d'apporter également une réponse à la question de l'enseignement de la pensée informatique. Ils examinent les effets de l'utilisation d'un robot programmable au sol, Bee-Bot, sur le développement de la

pensée informatique de 50 jeunes apprenants âgés entre cinq et six ans dont ils prennent en considération le genre.

Ils font l'hypothèse que l'étayage jouerait un rôle non négligeable dans le développement des compétences en pensée informatique pendant l'apprentissage avec Bee-Bot et ce par le fait que ce robot ne permette pas d'apporter de représentation visuelle de la démarche utilisée lors de la programmation. L'utilisation d'un système de mémoire externe permettrait aux apprenants de garder une trace visuelle des commandes afin de programmer le robot et viendrait ainsi soutenir l'apprentissage, notamment lors de l'utilisation de séquences d'instructions plus complexes contenant une diversité des commandes (par exemple, avancer, reculer, tourner à gauche, tourner à droite). De plus, ils s'interrogent sur le fait de savoir si les garçons et les filles peuvent bénéficier du même type d'étayage.

Ils s'attendent à ce que les notions « *avancer* » et « *reculer* » ne prêtent pas à confusion pour les apprenants puisqu'ils utilisent déjà ces termes dans la vie de tous les jours. De plus, ces notions sont identifiables par le biais de caractéristiques distinctes comme, par exemple, la partie avant ou arrière du visage d'une personne.

A contrario, ils supposent que ce ne sera pas le cas pour les notions « *gauche* » et « *droite* » puisqu'ils éprouvent davantage de difficultés à les utiliser de manière adéquate. La rotation imaginaire que les apprenants doivent effectuer de leur propre position ou de celle d'autrui est généralement possible entre 7 et 11 ans (Roberts & Aman, 1993 as cited by Angeli & Valadines, 2020).

Pour mener à bien leur étude, ils ont créé et implémenté trois scénarios pédagogiques autour d'un labyrinthe composé de 36 cases (6x6) et d'une histoire. Le premier scénario leur permet de faire connaissance avec le robot *Bee-Bot* et de découvrir ses différentes fonctionnalités. Les apprenants étaient ensuite amenés à réaliser plusieurs défis de programmation impliquant une combinaison des quatre directions variant de une à quatre commandes. Le second scénario est basé sur le même principe que son prédécesseur en termes d'ordre, de complexité et de durée mais ils y ajoutent la notion de contrainte. Enfin, le dernier scénario a été construit dans la continuité du précédent.

Ils ont également mis en place deux techniques d'étayage différentes lors du second scénario. Celles-ci ont été conçues en tenant compte des différences entre les deux genres. La première technique d'étayage permettait à l'apprenant d'utiliser des petites cartes de programmation plastifiées représentant chaque commande utiles afin de pouvoir former une séquence pour

résoudre chaque défi qui leur était posé. La deuxième technique d'étayage engageait chaque apprenant de manière individuelle en collaboration avec le chercheur. Face au même labyrinthe couplé à une représentation en trois dimensions du robot, les apprenants étaient amenés à réfléchir à la manière de résoudre le défi qui leur était proposé. Le chercheur, quant à lui, devait inscrire sur une feuille les commandes que l'apprenant lui dictait afin d'obtenir une représentation visuelle de l'algorithme créé avant de tester sa programmation sur *Bee-Bot*.

Suite à leur étude, ils font état de gains d'apprentissage statistiquement significatifs après l'implémentation de leur dispositif.

En outre, les garçons et les filles ayant bénéficié de techniques d'étayage différentes, les résultats montrent un effet d'interaction statistiquement significatif entre le genre et la stratégie d'étayage. En effet, les garçons ont bénéficié davantage de l'activité individualiste, kinesthésique, orientée vers l'espace et basée sur une manipulation avec des cartes de programmation, tandis que les filles ont bénéficié davantage de l'activité d'écriture collaborative (Angeli & Valanides, 2020).

Chou et Shih (2021), deux chercheurs taiwanais, ont mené une étude afin d'examiner si les stratégies d'apprentissages qu'ils ont mises en place avaient une influence sur le développement des compétences de base en pensée informatique. Trente-six élèves âgés de six ans y ont participé sur base volontaire. Ils ont reçu une formation de trois heures à la suite de laquelle ils ont été confrontés à un post-test composé de trois challenges de difficulté croissante catégorisés en fonction de la longueur de la séquence : facile (moins de 10 étapes), modéré (entre 11 et 20 étapes) et difficile (entre 21 et 30 étapes). Les chercheurs ont développé deux stratégies d'apprentissages différentes : premièrement, les élèves n'avaient recours à aucune aide visuelle. Après avoir examiné le labyrinthe, ils devaient programmer le robot. Deuxièmement, les élèves avaient recours à une aide visuelle grâce à l'utilisation de cartes de programmation qu'ils devaient placer devant eux avant de programmer le robot. Les résultats nous indiquent que l'utilisation de ces deux techniques d'étayage ne produit des résultats d'apprentissage que pour les tâches les plus compliquées. L'une des explications qu'ils avancent réside dans le fait que les élèves auraient besoin d'aide visuelle pour soutenir leur processus d'apprentissage lorsqu'ils sont confrontés à des problèmes qui nécessitent une réflexion de niveau supérieur. À l'inverse, l'aide visuelle ne serait pas nécessaire dans les défis de niveau inférieur. Néanmoins, les aides visuelles restent un moyen efficace pour aider les apprenants à s'engager dans les différents niveaux de pensée informatique (Chou & Shih, 2021).

Moore et al. (2020) ont mené une recherche dans des classes de deuxième année primaire et ont, quant à eux, utilisé le même robot pédagogique qui nous servira dans cette recherche. Ils font état d'un manque de fluidité au niveau des représentations dont les élèves ont besoin pour passer d'une forme à une autre, comme celle d'une représentation physique vers une autre plus abstraite. Ils constatent que les apprenants utilisent le langage ou encore la gestuelle pour s'aider à traduire les différentes représentations. Par ailleurs, les niveaux de développement de représentations des élèves évoluent en fonction de la difficulté de la tâche qui leur est proposée et les notions de structuration spatiale restent compliquées. La compréhension des stratégies utilisées pour soutenir les transitions au niveau des représentations pourrait aider les enseignants à inclure des techniques d'étayages susceptibles de permettre aux élèves de soulager, tout au moins en partie, la charge cognitive inhérente à ces traductions. De plus, ils émettent l'hypothèse que, plus leur expérience des représentations et des traductions de la pensée informatique augmentent, moins le recours aux représentations intermédiaires se ferait ressentir. Ceci indiquerait une meilleure acquisition de la fluidité des représentations (Moore et al, 2020).

Une étude réalisée par Nogry en 2018, s'est intéressée aux processus d'apprentissages mis en œuvre lors d'une séquence utilisant la robotique pédagogique chez des enfants âgés de 6 ans. Ceux-ci, répartis en triades, avaient pour objectif de programmer un robot de sol "*Bee-Bot*" dans une démarche d'investigation scindée en deux phases. La première visait l'exploration du matériel proposé tandis que la seconde reposait sur un étayage proposant l'aide d'une représentation graphique.

Lors de la phase de découverte, les apprenants se retrouvent face à certains obstacles tels "qu'une conception anthropomorphique du robot ou des difficultés à intégrer le caractère différé de l'exécution. [...] Ils rencontrent également une difficulté de compréhension des commandes du Bee-Bot" (Nogry, 2018, p.242).

Les échanges entre apprenants sont réduits à l'expression de sentiments tels que la joie ou le désappointement en fonction du résultat qu'ils obtiennent (Nogry, 2018).

Lors de la seconde phase, l'étayage mis en place focalise leur attention sur la production d'une procédure qu'il va d'abord falloir écrire. Ils acquièrent progressivement la notion de séquence d'instructions tout en façonnant leur démarche de résolution de problème (Nogry, 2018). L'auteure ajoute que « *les médiations proposées par l'enseignante ainsi que les représentations graphiques à élaborer jouent un rôle central dans cet apprentissage* » (Nogry, 2018, p.242).

2.6.3. Quelques pistes pour l'enseignement de la pensée informatique

Les résultats de ces différentes études dirigent les chercheurs vers davantage d'investigations en prenant en considération plusieurs pistes en matière de pensée informatique (Angeli & Giannakos, 2020) :

- la définition des compétences pour chaque niveau d'enseignement ou chaque niveau de développement des apprenants, afin d'apporter un socle de référence et un langage commun dans différents contextes ;
- l'utilisation de métaphores pour enseigner les concepts de manière efficace et efficiente et transformer les concepts abstraits en notions plus concrètes et plus simples à comprendre ;
- l'utilisation de stratégies et de technologies pédagogiques ;
- le développement professionnel des enseignants leur permettant de se préparer à l'enseignement de la pensée informatique et à intégrer des outils technologiques adéquats ;
- l'évaluation des compétences et des aptitudes en la matière, un domaine de recherche qui n'en est toujours qu'à ses débuts.

Angeli & Giannakos (2020) présentent enfin un plan en cinq étapes sur la façon dont ces cinq domaines de recherches peuvent être approchés lors de futures recherches (Fig.3). Ils le présentent de manière circulaire puisqu'ils comptent sur le fait que les progrès et la pratique relatifs à un domaine en particulier puissent en informer les autres et ainsi les faire évoluer à leur tour.

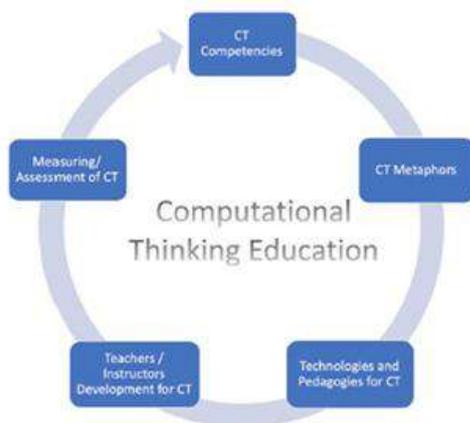


Figure 3 / A five-step research plan for CT Education (Angeli & Giannakos, p.2, 2020)

3. Question de recherche et hypothèses

Cette recherche poursuit deux objectifs. Le premier réside dans le fait de concevoir un scénario pédagogique qui permettrait de développer les compétences relatives à la pensée informatique et algorithmique (PIA) chez des élèves du premier cycle de l'enseignement primaire de la Fédération Wallonie-Bruxelles. Le second, dans la lignée des recherches menées par Angeli et Valanides en 2020, est d'y intégrer deux procédures d'étayage différentes, de voir si l'implémentation de l'une ou l'autre technique engendre une différence au niveau de l'acquisition des compétences en PIA et si cette différence se veut genrée.

Ceci nous amène donc à définir notre problématique de recherche comme suit :

« L'implémentation de scénarios pédagogiques couplée à deux techniques d'étayage différentes peut-elle faciliter l'acquisition de compétences spécifiques liées à la pensée informatique et algorithmique chez des filles et des garçons du premier cycle de l'enseignement fondamental de l'enseignement ordinaire en Fédération Wallonie-Bruxelles ? »

Au regard de notre littérature de recherche, nous émettons les hypothèses suivantes :

Hypothèse 1 : Les élèves progressent dans l'acquisition des compétences de pensée informatique et algorithmique suite à l'implémentation de scénarios pédagogiques spécifiques au développement de celles-ci.

Hypothèse 2 : L'acquisition de ces compétences se maintient dans le temps.

Hypothèse 3 : L'acquisition des compétences spécifiques à la PIA diffère selon la technique d'étayage mise en œuvre.

Hypothèse 4 : L'implémentation de ces deux techniques d'étayage va jouer un rôle différent sur le développement des compétences en PIA des élèves en fonction du genre.

4. Méthodologie

Tels qu'Angeli & Giannakos (2020) l'ont notamment souligné, la pensée informatique et l'algorithme ouvrent le champ à bon nombre de recherches. Dans ce cadre, nous allons nous intéresser plus amplement au domaine de l'utilisation de stratégies pédagogiques.

Afin de répondre à notre question de recherche et mettre nos hypothèses à l'épreuve, nous avons choisi de mettre en place une approche mixte recourant à différents outils que nous allons à présent vous présenter.

4.1. Description du contexte et de la population échantillonnée

Cette étude a été réalisée auprès d'un échantillon de convenance de 15 élèves scolarisés dans une classe de première et deuxième année de l'enseignement ordinaire communal en province de Luxembourg. Tous les élèves sont à l'heure.

L'indice socio-économique de l'établissement auquel appartient cette classe est de 16.

Aucun d'entre eux n'a eu auparavant d'expérience avec des activités relatives à la PIA, que ce soit à l'intérieur ou en-dehors de l'établissement scolaire qu'ils fréquentent et cette approche ne fait pas partie des pratiques pédagogiques courantes de l'enseignante.

Huit de ces élèves, quatre filles et quatre garçons, sont inscrits en première année primaire. Les sept autres, quatre filles et trois garçons, sont inscrits en deuxième année primaire.

Ces élèves ont été répartis aléatoirement en quatre sous-groupes, eux-mêmes subdivisés pour les besoins de certaines activités proposées, et ce, en fonction de l'année à laquelle ils appartiennent et suivant le schéma suivant (fig.4).

Dans le cadre de cette recherche, nous endosserons un double rôle à la fois de chercheuse puisque nous allons créer un outil, récolter des données et analyser ces dernières, et ensuite d'animatrice puisque nous nous chargerons de l'implémenter nous-même.

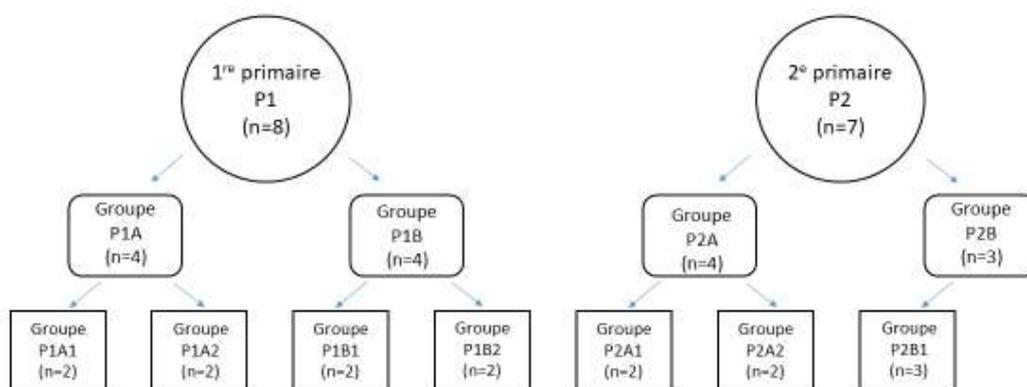


Figure 4 / Répartition de l'échantillon

4.2. Vigilances éthiques

Chaque parent ou tuteur légal des élèves de notre échantillon a été informé du fait que nous réalisons une étude portant sur le développement de la pensée informatique et algorithmique. Ces personnes ont reçu un document reprenant une brève description de notre recherche et une demande d'autorisation à la participation des enfants à celle-ci. Nous les avons informées de la libre participation de ces derniers ainsi que de la protection de leur anonymat et de la confidentialité des données récoltées qui ne feront en aucun cas l'objet d'une diffusion publique. Nous nous sommes tenues à disposition des personnes responsables pour toute question ou tout complément d'information.

Les prises de données nécessaires à cette recherche requièrent l'enregistrement audio et vidéo des enfants. Nous avons fourni à chaque personne responsable une autorisation de droit à l'image. Les demandes d'autorisations nous sont toutes revenues signées. S'il s'était avéré que l'une d'entre elles ait été refusée, nous aurions veillé à respecter leur choix et aurions pris soin de ne pas filmer le ou les enfant(s) en question et/ou serions restées attentives à la participation du ou des élève(s) dans les activités scolaires. Les formulaires distribués sont ceux rédigés par le comité éthique qui a approuvé préliminairement nos prises de données.

4.3. Schéma des différentes étapes de notre dispositif

La figure ci-dessous (fig.5) présente les différentes étapes successives de notre dispositif qui sera détaillée dans cette partie du travail.

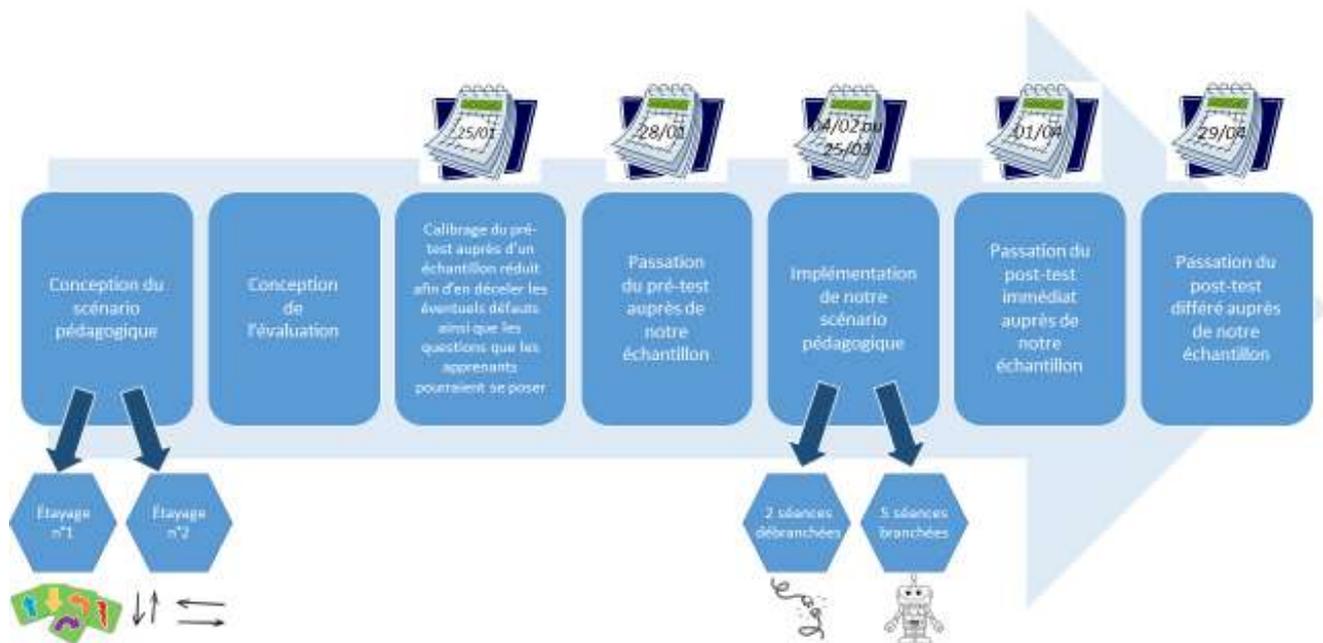


Figure 5 / Différentes étapes du dispositif

4.4. Conception d'un scénario pédagogique incluant deux techniques d'étayage

Afin de donner sens à notre question de recherche, nous avons été amenées à créer un scénario pédagogique dans lequel deux techniques d'étayage ont été mises en œuvre, et ce, afin d'en étudier les effets sur le développement des compétences en pensée informatique que nous avons ciblées. Nous allons vous présenter dans cette partie la description de ces techniques ainsi que du scénario.

4.4.1. Les techniques d'étayage

Le robot-souris que nous utilisons ne dispose d'aucune représentation visuelle des commandes que les apprenants utilisent lors de sa programmation et ce au même titre que le robot Bee-Bot utilisé dans la recherche menée par Angeli et Valadines (2020). Cette faiblesse inhérente au

robot-souris se couple au peu de mémoire de travail disponible chez les apprenants en lui imposant ainsi une charge cognitive externe importante telle que nous l'avons déjà souligné avec Sweller (1994) puisqu'ils doivent être capables de gérer toute une série de tâches simultanées. Suite à ce constat nous avons décidé de mettre en place deux techniques d'étayage en nous appuyant sur la recherche d'Angeli et Valadines (2020).

4.4.1.1. Première technique d'étayage (E1)

Le premier outil d'étayage a été conçu de façon à fournir à chaque apprenant des cartes de programmation reprenant les quatre commandes directionnelles. Dans le cadre des activités débranchées, nous avons conçu des cartes au format A5 (fig.6) sur lesquelles sont représentées les flèches dont les couleurs s'associent à celles présentées sur le dos de la souris-robot que nous utiliserons ultérieurement. De plus, nous avons représenté un repère dans le coin inférieur droit afin d'en indiquer le sens adéquat.

Lors des activités branchées, nous avons fourni aux élèves les petites cartes de programmation présentes dans le kit d'activités proposé par "Learning Resources" (fig.7). Celles-ci représentent chacune des commandes présentes sur le dos du robot. Sur un côté de la carte (fig.8), nous retrouvons uniquement la flèche tandis que sur l'autre côté de la carte (fig.9), nous pouvons y voir en plus la représentation de la souris et du sens de cette dernière.

Tout au long de notre scénario pédagogique, l'élève est ainsi invité à placer les cartes dans l'ordre qu'il pense être le plus adéquat afin de former l'algorithme dont il a besoin pour résoudre le problème auquel il est confronté. Nous avons pris le parti de ne pas induire de sens d'écriture aux élèves pour leur laisser la possibilité de placer les cartes comme bon leur semble et ainsi pouvoir également en discuter avec leurs camarades.

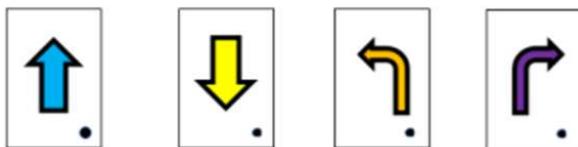


Figure 6 / Fiches au format A5



Figure 7 / Cartes de programmation présentes dans le kit "Learning Resources"



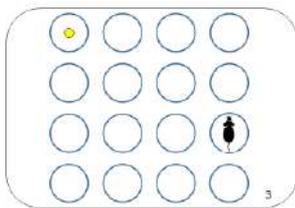
Figure 8 / Flèche "avance"



Figure 9 / Flèche "avance" accompagnée de la représentation de la souris-robot

4.4.1.2. Seconde technique d'étayage (E2)

Le second outil d'étayage a été conçu de telle sorte à permettre aux apprenants d'écrire leur algorithme avant de pouvoir le tester. Ils pouvaient disposer de tout le matériel d'écriture nécessaire, en ce compris des crayons ou marqueurs de couleur.



Dans le cadre des activités débranchées, nous leur avons fourni la représentation du défi à réaliser (fig.10) ainsi qu'une feuille de papier au format A4.

Figure 10 / Représentation du défi à réaliser

Lors des activités branchées, nous avons fourni à chaque apprenant la représentation du plateau de jeu, de la souris-robot ainsi que de ses commandes en taille réduite et imprimée sur une feuille au format A4 (fig.11). En-dessous de chacune d'entre elles, l'élève a la possibilité d'écrire les représentations des actions dont il a besoin pour construire son algorithme afin de programmer son robot et l'amener à réaliser la tâche qui lui est demandée.

Nous avons choisi de ne pas ajouter d'espace prédéfini à l'écriture des commandes pour permettre aux élèves d'utiliser l'espace dont ils ont besoin, ce y compris la représentation du labyrinthe.

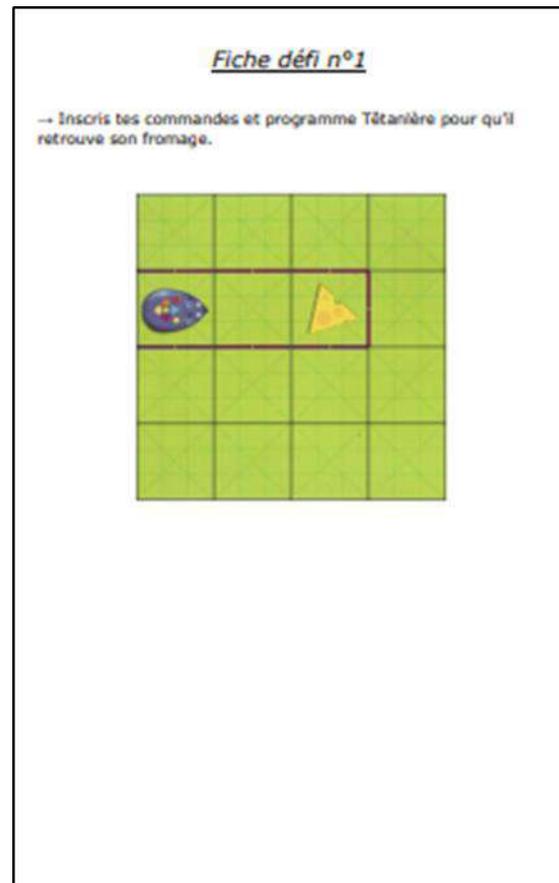


Figure 11 / Fiche défi

4.4.2. Le scénario pédagogique

Notre recherche s’articule autour d’un scénario pédagogique (cf. annexe 2, p. 106) que nous avons élaboré au départ de l’album de jeunesse « Les aventures de Pensatou et Têtanlère, le château de Radégou » afin de donner du sens à nos activités. Celles-ci sont conçues au départ du canevas finalisé par le CRIFA (Denis et Vandepuit, 2006) et exploité dans le cadre du projet ERASMUS+ PIAF (Parmentier et al., 2000).

Ce dernier renferme douze rubriques, à savoir : le titre, les informations pratiques, la description, les compétences et objectifs spécifiques à la PIA, les prérequis à l’activité, les ressources numériques, l’organisation de la classe, le déroulement du scénario et de ses activités, l’évaluation, le retour réflexif sur le scénario construit, la bibliographie, et, enfin, les annexes éventuelles.

Notre scénario dénombre 7 séances (N=7), dont deux de type débranchées et cinq de type branchées, d’environ une période (soit cinquante minutes) que nous allons détailler. Dans le tableau ci-dessous (tableau 1), vous trouverez un aperçu global des sous-compétences travaillées au cours des différentes séances.

Tableau 1 / Sous-compétences cibles en fonction des séances

Séance	Sous-compétences travaillées*						
	1.5. Prédire le résultat d'une séquence d'actions	2.3. Créer une séquence d'actions pour atteindre un objectif simple	2.4. Créer une séquence d'actions pour atteindre un objectif complexe	3.1. Répéter une séquence d'actions un nombre donné de fois	6.1. Vérifier si une séquence d'actions atteint un objectif donné	6.2. Repérer des erreurs dans une séquence d'actions	6.3. Corriger une séquence d'actions pour atteindre un objectif donné
1		X			X		
2		X			X		
3		X			X	X	X
4		X			X	X	X
5		X	X	X	X	X	X
6		X	X	X	X	X	X
7	X	X	X	X	X	X	X

(*) Ces sous-compétences sont issues du référentiel développé dans le cadre du projet Erasmus+ PIAF

4.4.2.1. Les activités débranchées

a) Première séance

La première activité intitulée « *(re)découverte de l'album* » consiste à faire (re)découvrir aux apprenants l'album de jeunesse « Tarr, L., & Devaux, M. (2016). Les aventures de Pensatou et Têtanlère, le château de Radégou. Editions EP&S ».

La deuxième activité s'intitule « *Aide Pensatou à rejoindre le château de Radégou en passant par le vieux garage* ». Après avoir organisé l'espace, l'enseignant(e) place deux cerceaux au sol dans lesquels elle place deux images. Ces images représentent « la maison des souris » ainsi que « le vieux garage ». Après avoir clarifié avec les élèves les notions de déplacement, les élèves doivent, dans un premier temps, donner des instructions à l'enseignante (dans le rôle de Pensatou) pour que celle-ci se déplace du point d'origine (la maison des souris) à la destination finale (le vieux garage). Dans un deuxième temps, les élèves prennent tour à tour le rôle de Pensatou. Enfin, pour terminer, l'enseignante leur propose de représenter eux-mêmes la séquence d'actions au tableau avant de donner les instructions à Pensatou.

b) Deuxième séance

Lors de **la troisième activité** intitulée « *Récupère les perles dans le vieux garage* » les élèves, répartis en sous-groupe de maximum quatre, sont amenés à récupérer une perle dans le vieux garage. Pour ce faire, l'enseignante dessine un quadrillage au sol (4x4) et distribue des fiches (N=12) aux élèves (fig.12). On y retrouve le quadrillage représenté au sol, une souris et une perle. Après avoir repéré l'emplacement de chacun de ceux-ci, les élèves sont amenés à construire la séquence d'actions qui permettra à Pensatou de récupérer sa perle dans le vieux garage ainsi qu'à la tester à l'aide d'un élève. Cette activité se différencie en fonction de la technique d'étayage qui leur est attribuée (cf. supra). Pour la première technique d'étayage, les élèves reçoivent une feuille vierge et du nécessaire d'écriture. Pour la seconde technique d'étayage, ils reçoivent des flèches au format A5 dont 10 bleues, 4 jaunes, 4 oranges et 4 mauves. Les couleurs correspondent aux commandes de la souris-robot que nous utilisons par la suite (fig.13).

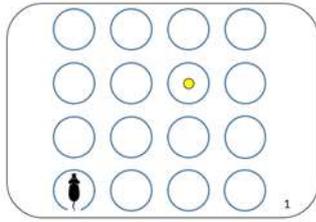


Figure 12 / Exemple de fiche activité 3

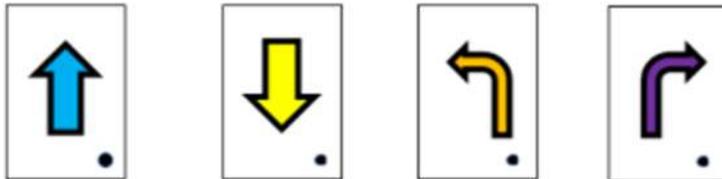


Figure 13 / Flèches format A5

4.4.2.2. Les activités branchées

Étant donné le jeune âge de nos participants et pour faire sens à l'album de jeunesse qu'ils ont découvert en amont, nous avons décidé d'utiliser le robot-souris Code & Go® (fig.14) présent dans le set d'activités proposé par « Learning Resources » (fig.15) que nous utilisons également.



Figure 14 / Robot-souris Code & Go



Figure 15 / Kit d'activités « Learning Resources »

Ce set comprend :

- Une souris-robot,
- Un morceau de fromage,
- 30 cartes de programmation codées par couleur correspondant aux boutons de la souris.
- 22 murs de labyrinthe,
- 16 plaques de labyrinthe qui se raccordent pour former un grand plateau,
- 3 tunnels.
- 10 cartes d'activités de labyrinthes recto-verso.

Pour les besoins du scénario pédagogique, nous avons également créé :

- Des nouvelles cartes de programmation « X2 » et « X3 » afin de pouvoir intégrer la notion de boucle (fig.16).

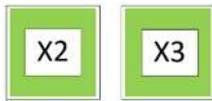


Figure 16 / Cartes de programmation X2 / X3

- 6 cartes d'activités de labyrinthes avec une perle (cartes 21 à 26) ainsi que 5 cartes d'activités de labyrinthes avec une perle et un Radégou, personnage principal de l'album de jeunesse (cartes 27 à 31) afin de pouvoir intégrer la notion de contrainte.
- 31 fiches au format A4 reprenant sur chacune d'elles une représentation de chaque carte-défi afin de pouvoir mettre en place la première technique d'étayage.

Les élèves sont répartis en dyade ou en triade tout au long des activités branchées. Celles-ci restent identiques en fonction de la répartition aléatoire qui a été effectuée.

a) Troisième séance

Lors de **la quatrième activité** intitulée « *Faisons connaissance avec le robot Têtanlère* » les élèves, répartis par groupe de trois ou quatre élèves, reçoivent deux souris-robots. L'enseignante laisse un temps d'appropriation aux élèves afin de les laisser découvrir l'objet et identifier la couleur et la fonction de chacun des boutons présents sur le dos de la souris. Ensuite, elle propose aux élèves une représentation (A3) (fig.17) de la souris ainsi que des étiquettes (munies de velcro) relatives aux fonctions de la souris (fig.18). En grand groupe, ceux-ci doivent verbaliser leurs découvertes et replacer les actions au bon endroit. Ensuite, ils repartent en sous-groupes pour s'approprier à nouveau la souris-robot en fonction des découvertes qui ont été faites. Notons que cette synthèse restera à disposition des élèves tout au long de leurs futures activités.

Les fonctions de la souris-robot

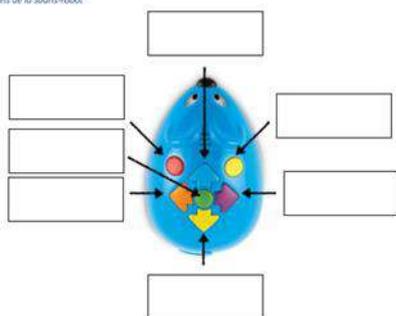


Figure 17 / Représentation de la souris



Figure 18 / Étiquettes fonctions

b) Quatrième séance

Lors de **la cinquième activité** intitulée « *Le robot Têtanlère récupère le fromage* », les élèves, répartis en dyade ou en triade, reçoivent un set de jeu complet (tel que décrit plus haut) et, en fonction de la technique d'étayage qui leur a été attribuée, soit des petites cartes de programmation, soit les fiches défi au format A4. L'enseignante leur distribue ensuite les cartes d'activités de difficulté croissante en commençant par la première (fig.19). Le groupe assemble les différentes pièces du labyrinthe pour reformer le plateau de jeu qu'il peut voir sur sa carte et place la souris-robot et le fromage comme indiqué pour ensuite le programmer pour qu'il récupère son fromage. Ils évoluent à leur rythme et reçoivent une nouvelle carte dès qu'ils ont réussi la précédente.

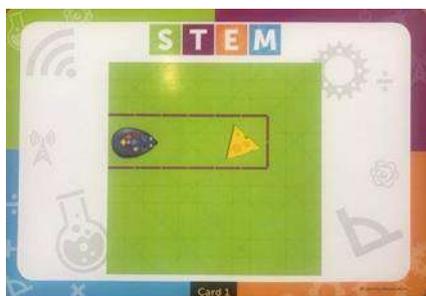


Figure 19 / Carte d'activité

c) Cinquième séance

La sixième activité s'intitule « *Le robot Têtanlère récupère le fromage plus vite, plus vite...* ». Après avoir introduit la notion de boucle au départ de la fiche d'activité numéro 5 (fig.20), les élèves reçoivent, soit la représentation graphique de ces nouvelles commandes (étayage n°1), soit des nouvelles cartes de programmation X2 et X3 (étayage n°2).

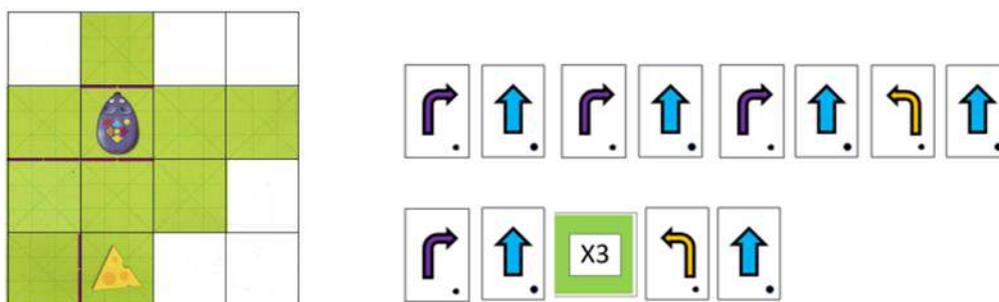


Figure 20 / Fiche d'activité n°5 accompagnée des commandes

Ils évoluent ensuite à leur rythme en poursuivant avec les fiches d'activités qui suivent celles auxquelles ils étaient arrivés lors de la quatrième séance.

d) Sixième séance

La septième activité s'intitule « *Le robot Têtanlère récupère le fromage en passant chercher une perle au garage et sans se faire attraper par Radégou* ». Toujours selon la même organisation que les séances précédentes, l'enseignante introduit à présent une contrainte (la perle ou Radégou) et distribue aux élèves les fiches d'activités 21 à 31 qu'ils découvrent et travaillent en autonomie l'une après l'autre dès qu'ils réussissent leur défi. Ils continuent néanmoins d'utiliser la notion de boucle vue lors de la séance précédente.

e) Septième séance

Lors de **la huitième activité** intitulée « *Où se cache le fromage ?* », les élèves vont utiliser tout le matériel mis à leur disposition pour créer un parcours selon leurs envies. Un des élèves place la souris sur le plateau de jeu et va coder une séquence (à l'aide des cartes de programmation ou en l'inscrivant sur une feuille). L'autre doit prédire le déplacement de la souris et placer le fromage sur le plateau. Ensemble, ils programment le robot pour vérifier que le morceau de fromage est placé correctement.

4.5. Recueil et analyse de données

Cette recherche s'appuie sur une méthode mixte recourant à différents outils afin de vérifier les hypothèses qui ont été formulées.

Dans le but de vérifier la première hypothèse (H1) concernant la progression des élèves dans l'acquisition des compétences de pensée informatique et algorithmique suite à l'implémentation de scénarios pédagogiques spécifiques au développement de celles-ci, nous procéderons à la

comparaison des résultats entre le pré-test et le post-test immédiat ainsi qu'à l'analyse des enregistrements vidéos des activités menées par l'intermédiaire de la grille d'observation.

La deuxième hypothèse (H2) concerne quant à elle le maintien de ces compétences dans le temps qui sera mesuré en comparant les résultats entre le post-test immédiat et le post-test différé effectué quatre semaines après l'implémentation de l'entièreté du scénario pédagogique.

Pour éprouver les troisième (H3) et quatrième (H4) hypothèses consistant à analyser la différence éventuelle dans le développement des compétences en PIA en fonction de la technique d'étayage qui aura été mise en place, et également en fonction du genre auquel appartiennent les apprenants, nous procéderons à la comparaison des résultats entre les différents tests ainsi qu'à l'analyse des occurrences des conduites témoins de l'acquisition des compétences.

Afin de vérifier si les différences que nous aurons observées entre les moyennes sont statistiquement significatives, nous procéderons à une analyse statistique *t* de Student grâce au logiciels « SAS studio » et « Excel ».

Enfin, nous consigneront pendant ou juste après chaque séance nos observations personnelles dans un journal de bord (cf. annexe 3, p. 183).

4.5.1. L'évaluation des compétences en PIA

4.5.1.1. *Pré-test, post-test immédiat et post-test différé*

a) Conception du test

Notre évaluation (cf. annexe 4, p. 298) conçue au format papier se retrouve implémentée à trois moments distincts de notre dispositif : avant toute intervention, directement suite aux séances reprises dans notre scénario pédagogique et, enfin, un mois après la fin de ce dernier.

Nous avons construit notre évaluation en partie sur base du BCTt conçu par Zapata-Cáceres et ses confrères (2020). En effet, nous avons sélectionné les items pertinents (items de 1 à 10) à utiliser au regard du référentiel de compétences que nous utilisons ainsi que du scénario pédagogique que nous avons créé. Nous y avons ajouté 13 autres items que nous avons construits nous-même. Ci-dessous, vous retrouvez le tableau récapitulatif des différents items au regard du format de la question, de la taille du labyrinthe (9 ou 16 cases), de la présence ou non d'un obstacle à éviter (un chat), du ramassage ou non d'un objet (une fleur) sur le chemin à parcourir ainsi que de la ou des compétence(s) visée(s) (cf. tableau 2).

Tableau 2 / Tableau récapitulatif des différents items de notre évaluation

Item	Format de la question	Taille du labyrinthe	Obstacles	Ramassage	Compétences PIA					
					1. Définir des abstractions / généraliser	2. Composer / décomposer une séquence d'actions	3. Contrôler une séquence d'actions	4. Evaluer des objets ou des séquences d'actions	5. Manipuler des représentations formelles	6. Construire une séquence d'actions de manière itérative
1	QCM	Petit								6.1
2	QCM	Petit	X							6.1
3	QCM	Petit	X	X						6.1
4	QCM	Grand	X	X						6.1
5	QCM	Petit				2.4	3.1			6.1
6	QCM	Petit				2.4	3.1			6.1
7	QCM	Petit	X			2.4	3.1			6.1
8	QCM	Grand				2.4	3.1			6.1
9	QCM	Grand		X		2.4	3.1			6.1
10	QCM	Grand				2.4	3.1			6.1
11	Cocher une case	Petit			1.5					
12	Cocher une case	Petit			1.5					
13	Cocher une case	Grand			1.5					
14	Entourer et écrire les commandes	Petit								6.2/6.3
15	Entourer et écrire les commandes	Grand								6.2/6.3
16	Ordonner et écrire les commandes	Petit				2.1				
17	Ordonner et écrire les commandes	Grand	X			2.1				
18	Compléter et écrire la commande	Petit				2.2				
19	Compléter et écrire les commandes	Grand		X		2.2				
20	Écrire les commandes	Petit				2.3				
21	Écrire les commandes	Petit		X		2.3				
22	Écrire les commandes	Grand	X			2.3				
23	Écrire les commandes	Grand	X	X		2.3				

Tous les items (fig.21 & 22) sont construits avec un labyrinthe (pouvant être de petit ou de grand format), l'objectif est de ramener le poussin jusqu'à sa maman avec parfois un obstacle à éviter (le chat) ou un objet à ramasser (une fleur). Chaque item est construit avec le moins de texte possible et illustré par des symboles présumés assez explicites afin de permettre aux plus jeunes n'ayant pas encore acquis toutes les compétences en lecture de pouvoir comprendre la tâche qui leur est demandée. Le nombre de propositions de réponses est porté à quatre afin de réduire la probabilité de réponse au hasard. Chaque réponse, quant à elle, est proposée verticalement de haut en bas ceci pour faire le lien avec le sens de lecture des séquences de code (Zapata-Cáceres et al., 2020). Enfin, nous avons également construit le correctif du questionnaire (cf. annexe 5, p.323).

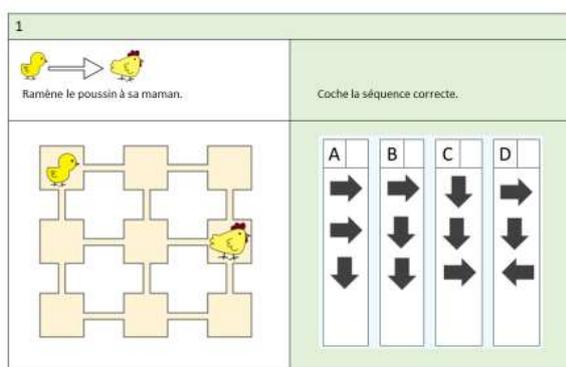


Figure 21 / Exemple d'item (1)

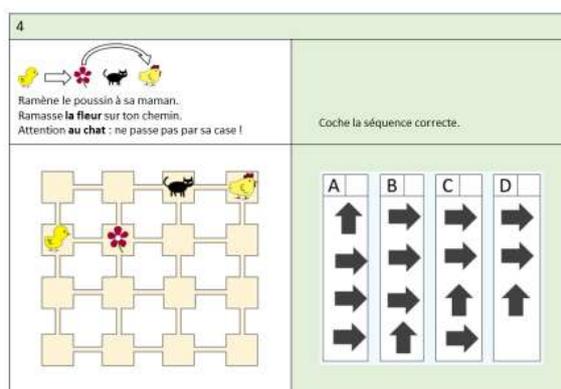


Figure 22 / Exemple d'item (2)

b) Calibrage du test

Cet outil a tout d'abord été pré-testé auprès de cinq élèves d'âge comparable à ceux de notre échantillon. Suite à ce calibrage, nous avons apporté deux brèves modifications :

- Aux questions n°14 et n°15, certains élèves demandent s'ils doivent réécrire toutes les commandes. Nous avons ajouté l'information dans la consigne « ... Réécris correctement toutes les commandes ».
- Aux questions n°16 et n°17, les élèves ne comprennent pas la consigne « ordonnez ». Nous avons remplacé la consigne par « mets la séquence dans l'ordre... ».

Les élèves ont également montré quelques problèmes de compréhension des consignes que nous avons anticipés lors de notre expérimentation :

- Certains élèves accordent plus d'importance à reproduire les grosses flèches telles que représentées dans le test plutôt qu'à l'exercice en lui-même. Nous avons rajouté des flèches moins épaisses et avons insisté sur leur représentation dans l'explication des consignes.
- Certains élèves ont éprouvé des difficultés avec le sens de la lecture du code (de haut en bas). Nous avons pris plus de temps à l'explication du sens de lecture avant de démarrer le test.
- Certains élèves ont demandé s'ils pouvaient dessiner sur le labyrinthe. Nous avons explicité aux élèves qu'ils avaient le droit de dessiner sur le quadrillage pour y représenter le chemin à parcourir ou pour y inscrire les flèches pour les aider.

c) Passation du test

Le test s'est déroulé de manière individuelle. Chaque élève en a reçu un exemplaire sur lequel figurait le code anonymisé qui lui avait été accordé au préalable.

L'enseignante a d'abord expliqué les commandes (fig.23) et montré un exemple de question animé par l'intermédiaire d'un logiciel de présentation de diapositives (fig.24) en projetant l'ensemble sur le tableau blanc interactif.

L'enseignante a présenté et lu chaque question les unes après les autres en prenant soin de laisser le temps nécessaire aux élèves de répondre aux questions.

Dans son ensemble, explications comprises, le test a duré 50 minutes.

Explication des commandes

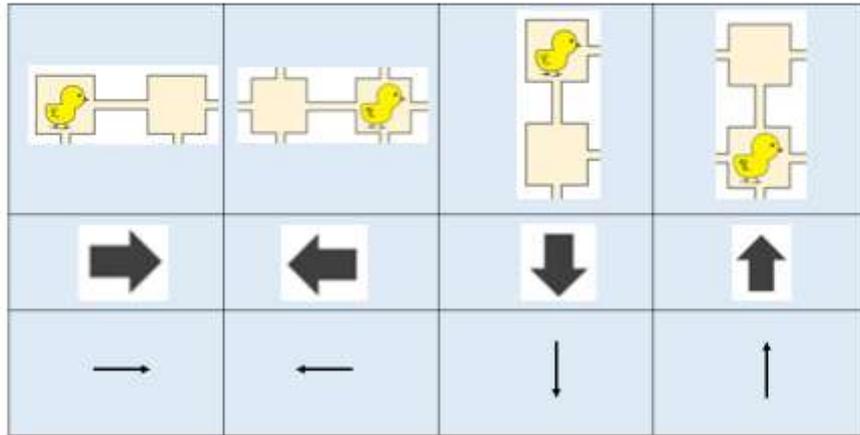


Figure 23 / Explication des commandes

Exemple de question

Le numéro de la question

1

Ramène le poussin à sa maman.

Coche la séquence correcte.

Le labyrinthe

Le poussin

La maman

Les consignes

La réponse

Les commandes

A	B	C	D

Figure 24 / Exemple de question animée

d) Analyse des données

Pour chaque élève, nous calculons les résultats obtenus au pré-test, au post-test immédiat ainsi qu'au post-test différé. L'objectif étant de pouvoir calculer un gain brut (GB), un gain possible (GP) ainsi qu'un gain relatif (GR).

Le gain brut correspond à ce qui a été réellement gagné et s'obtient en calculant la différence entre le résultat obtenu au post-test immédiat ou encore au post-test différé et celui obtenu au pré-test.

Le gain possible quant à lui s'obtient en soustrayant les résultats obtenus au pré-test au score maximum qu'il est possible d'obtenir aux post-tests.

Le gain relatif (exprimé en pourcents) est le rapport entre ce qui a été gagné et ce qui pouvait l'être, en l'occurrence le gain possible. Il se calcule en utilisant la formule suivante :

$$\text{GR} = \frac{\text{Score après - score avant}}{\text{Score maximum - score avant}} * 100 = \frac{\text{GB}}{\text{GP}} * 100$$

Si le résultat du pré-test s'avère être supérieur à celui du post-test, la formule du gain relatif ne peut pas être utilisée. Il convient alors de calculer une perte relative suivant la formule :

$$\text{PR} = \frac{\text{Score après - score avant}}{\text{Score avant}} * 100$$

4.5.1.2. La grille d'observation

L'observation des conduites des élèves et des interventions de l'animatrice durant les différentes étapes du scénario pédagogique s'est basée sur le codage de l'émission de conduites-témoins de la mise en œuvre de comportements (verbaux ou non verbaux) liés aux compétences en PIA ciblées dans le cadre de cette étude.

Les séances d'enseignement incluant l'une ou l'autre technique d'étayage (séances 2,4,5,6 et 7) ont été enregistrées et analysées à l'aide d'une grille d'observation (cf. annexe 6, p.326) également développée par le CRIFA dans le cadre du projet Erasmus+ PIAF et dont voici ci-dessous un extrait (fig.25).

Elle permet de distinguer, d'une part, les conduites des apprenants révélatrices de l'acquisition d'une ou de plusieurs compétences travaillées au cours des activités pédagogiques, et, d'autre part, les conduites de l'enseignant à l'initiative de celles-ci.

Cette grille reprend les six compétences principales ainsi que les vingt-six sous-compétences qui leur sont associées et qui découlent du référentiel de compétences élaboré dans le cadre du projet Erasmus+ PIAF. Elle est accompagnée de critères observables pour chaque sous-compétence qui la compose.

GRILLE D'OBSERVATION - Apprenants/Facilitateurs			Facilitateur Sollicite	Facilitateur Autres (par exemple action sur le matériel)	Apprenant 1	Apprenant 2
Définir des abstractions / généraliser	C.1.1	Nomme	Une action			
			Un objet			
			Une séquence d'actions			
	C.1.2	Différencie	Objet et action			
			Action atomique et non-atomique			
	C.1.3	Identifie	L' ou les objet(s) et/ou la ou les séquences d'actions dont il a besoin pour exécuter la séquence d'actions			
	C.1.4	Décrit	Le résultat final d'une séquence d'actions			
			Les étapes de la séquence d'actions exécutée			
	C.1.5	Prédit	Le résultat final d'une séquence d'actions			
			Les étapes de la séquence d'actions non exécutée			
C.1.6	Utilise	Une variable dont la valeur peut changer				
C.1.7	Reconnait	Les objets réutilisables pour atteindre un nouvel objectif				
		Les actions réutilisables pour atteindre un nouvel objectif				
		Les séquences d'actions réutilisables pour atteindre un nouvel objectif				
Composer/décomposer une séquence d'actions	C.2.1	Ordonne	Des actions pour atteindre l'objectif visé			
			Des séquences d'actions pour atteindre l'objectif visé			
	C.2.2	Complète	Une séquence d'actions par des actions fournies pour atteindre l'objectif			
			Une séquence d'actions par des actions non-fournies pour atteindre l'objectif			
	C.2.3	Crée	Une séquence d'actions pour atteindre un objectif simple			
	C.2.4	Crée	Une séquence d'actions pour atteindre un objectif complexe en utilisant la boucle (B) et/ou la condition (C)			
	C.2.5	Combine	Des séquences d'actions existantes pour atteindre un objectif			
C.2.6	Décompose	Une tâche en sous-tâches plus simple				
C.3.1	Répète	L' ou les action(s) un nombre donné de fois				

Figure 25 / Extrait de la grille d'observation

À chaque observation d'un comportement, nous inscrivons une unité (I) dans la case correspondante. De plus, nous utilisons la lettre qui permet de distinguer les comportements verbaux (V) des actions sur le matériel didactique (D) ou sur le logiciel (L). Nous notons ensuite entre parenthèses le nombre total d'unités dans chaque catégorie (fig.26).

GRILLE D'OBSERVATION - Apprenants/Facilitateurs			Facilitateur Sollicite	Facilitateur Autres (par exemple action sur le matériel)	Apprenant 1 / E15	Apprenant 2 / E5
Manipuler des représentations formelles	C.3.3	Améliore	Une séquence d'actions pour répondre à un critère donné Des objets au moyen d'une représentation formelle			
	C.3.1	Représente	Des actions au moyen d'une représentation formelle Des séquences d'actions au moyen d'une représentation formelle			
	C.3.2	Traduit	Des objets à l'aide d'une représentation formelle autre que celle d'origine			
			Des actions à l'aide d'une représentation formelle autre que celle d'origine			
			Des séquences d'actions à l'aide d'une représentation formelle autre que celle d'origine			
	Construire une séquence d'actions de manière itérative	C.6.1	Vérifie	Si une séquence d'actions atteint un objectif donné	11 (2)	LLLLLLLLLL (12)
C.6.2		Repère	Des erreurs dans une séquence d'actions		DDDD (4)	DD (2)
C.6.3		Corrige	Une séquence d'actions pour atteindre un objectif donné		DDDD (4)	DD (2)
C.6.4		Étend/Adapte ou réutilise	Une séquence d'actions pour atteindre un nouvel objectif			
			Verbalisation (V) Action sur le logiciel (L) Action sur matériel didactique (D)	 	  	

Figure 26 / Exemple de codage de la grille d'observation

4.5.1.3. Analyses statistiques

De manière à déterminer la significativité des différences de résultats observées, nous avons procédé à une analyse *t* de Student pour échantillons appariés avec un seuil de significativité α de 0,05, soit un indice de confiance à 95%. Nous avons codé les données de la manière suivante : 0 pour le premier étayage (cartes) et 1 pour le second étayage (écriture) ; 0 pour les garçons et 1 pour les filles.

Nous avons ensuite émis l'hypothèse nulle (h_0) selon laquelle la différence des moyennes était égale à zéro et donc, que les deux populations étaient identiques et qu'il n'existait pas de différence statistiquement significative entre elles. Dans le cas où la valeur *p* est égale ou se situe en-dessous du seuil de signification, nous pouvons rejeter l'hypothèse nulle et conclure que la différence entre les moyennes est statistiquement significative. Dans le cas contraire, si la valeur de *p* est supérieure au seuil de signification, nous ne pouvons pas rejeter l'hypothèse nulle et donc nous pouvons conclure que la différence entre les moyennes n'est pas statistiquement significative.

4.6. Limites méthodologiques

Les choix méthodologiques que nous avons opérés dans le cadre de notre recherche entraînent certaines limites d'un point de vue méthodologique.

Premièrement, l'absence d'un groupe contrôle ne nous permet pas d'infirmier avec certitude toute autre influence qui permettrait d'attribuer l'effet sur les compétences en PIA à la seule intégration de notre scénario pédagogique.

Deuxièmement, la taille réduite de notre échantillon de convenance ne nous permet pas d'inférer nos conclusions à l'ensemble de la population puisque ce dernier n'est pas représentatif. Il faut donc faire preuve d'une grande prudence quant aux conclusions que nous pourrions en tirer.

Troisièmement, la présence de la caméra pour entraîner un « effet d'intrusion de l'observateur » (Beaugrand, 1988, p.26) pouvant engendrer une modification du comportement de l'enseignant ou des apprenants.

Quatrièmement, l'usage de la grille d'observation peut être soumis à la propre interprétation de l'observateur.

Enfin, dans la construction de notre évaluation, le nombre d'items mesurant chaque compétence était restreint. Celle-ci ne nous permet pas de juger à elle seule de l'acquisition des compétences.

5. Présentation des résultats et interprétation

Dans cette partie, nous allons tout d'abord nous pencher sur l'analyse de la réussite des différents items de l'évaluation. Ensuite, nous mettrons en lumière les résultats de notre recherche que nous interpréterons hypothèse par hypothèse.

5.1. Analyse de la réussite des items de l'évaluation

Sur l'ensemble de l'évaluation, nous nous sommes intéressées à la moyenne de réussite des items au pré-test, au post-test immédiat et au post-test différé (fig.27).



Celle-ci est de 78% de réussite au pré-test contre 91% de réussite au post-test immédiat, soit une amélioration de 13%. Elle passe à 93% au post-test différé, soit une nouvelle augmentation de 2% par rapport au post-test immédiat.

Nous avons également calculé le pourcentage de réussite obtenu pour chaque item au pré-test, au post-test immédiat et au

Figure 27 / Pourcentage moyen de réussite pour chaque item

post-test différé.

Sur les 23 items du test, 16 ont été mieux réussis au post-test immédiat, 6 items obtiennent le même score et un seul item (I20) a été moins bien réussi.

Alors que lors du pré-test, seuls trois items (I11, I12 et I20) ont été réussis par tous les élèves, ce chiffre se porte à neuf lors du post-test immédiat.

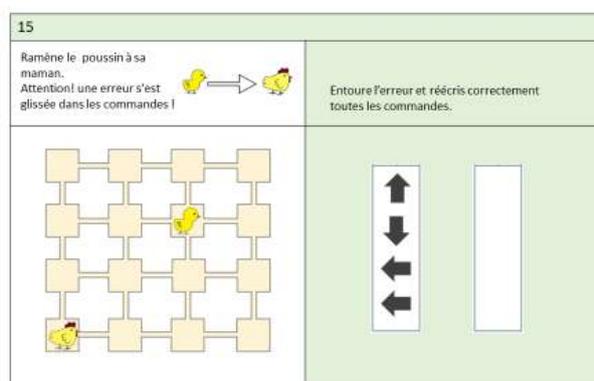


Figure 28 / Item n°15

L'item 15 (fig.28) est quant à lui celui qui a été le moins bien réussi sur l'ensemble du test, que ce soit lors du pré-test (20% de réussite) ou du post-test immédiat (66% de réussite). Le fait d'y avoir inclus deux consignes a peut-être été l'élément perturbateur de cet item. Il conviendrait de construire deux items au départ de celui-ci pour vérifier notre hypothèse.

Les items 11 (fig.29) et 12 (fig.30) ont été réussis par tous les élèves lors des deux évaluations.

Ces items sont les suivants :

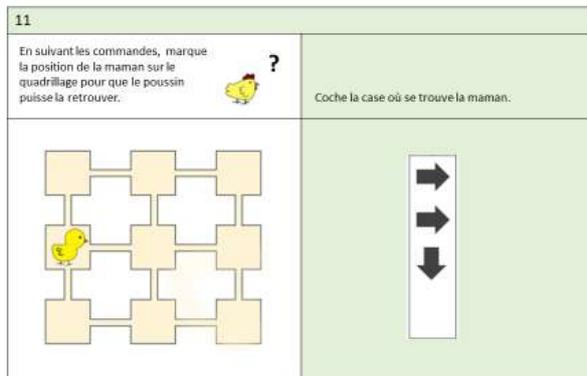


Figure 29 / Item n°11

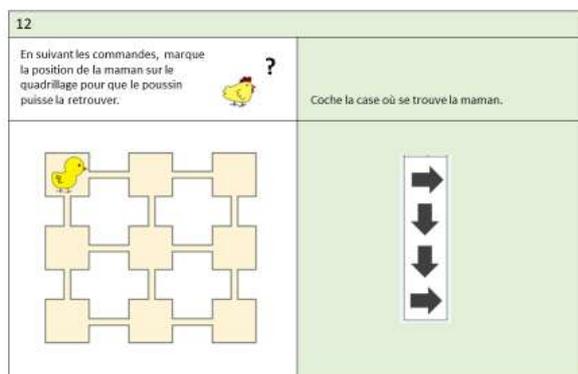


Figure 30 / Item n°12

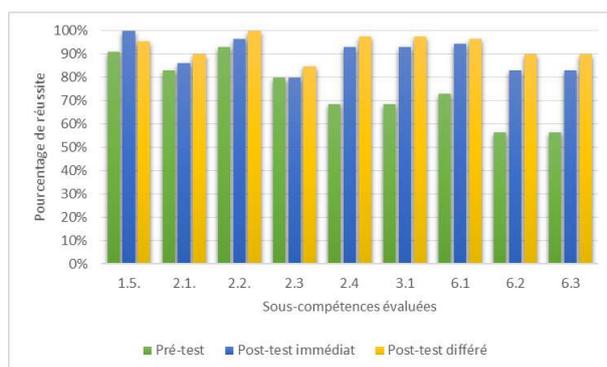


Figure 31 / Pourcentage de réussite en fonction des sous-compétences visées

compétences 6.2 Repérer des erreurs dans une séquence d'actions et 6.3 Corriger une séquence d'actions pour atteindre un objectif donné que nous observons la meilleure progression avec un pourcentage de réussite passant de 57% à 90%, soit une augmentation de 33%.

Ces deux items évaluent la compétence 1.5. "Prédire le résultat d'une séquence d'actions". Peut-être devrions nous nous interroger sur le caractère trop simple de ces deux questions ? Ceux-ci mériteraient d'être réévalués afin d'assurer leur validité. Ou alors, pourrions-nous émettre l'hypothèse que ces compétences sont également travaillées en classe dans d'autres disciplines ?

Nous nous sommes également intéressées au pourcentage de réussite en fonction des sous-compétences visées par l'évaluation. Dans la figure 31, nous pouvons constater premièrement que tous les élèves obtiennent au moins 50% de réussite. Deuxièmement, une augmentation de ce pourcentage entre le pré-test et le post-test immédiat, et ce, pour l'ensemble des sous-compétences visées et, troisièmement, que la compétence 1.5 « Prédire le résultat d'une séquence d'actions », est la seule à voir son pourcentage de réussite légèrement diminuer lors du post-test différé. Notons tout de même que cette baisse est de l'ordre de 5%. Mesurée à travers les items 11, 12 et 13, seul ce dernier a été échoué par deux élèves. Ce sont pour les

5.2. Acquisition des compétences en PIA (H1)

Nous allons tout d'abord vous présenter les résultats (tableau 3) obtenus suite à la passation des pré-test, post-test immédiat et post-test différé auprès des quinze élèves de notre échantillon. Nous les analyserons en fonction des erreurs qui ont été commises chez les élèves présentant une perte relative entre le pré-test et le post-test immédiat pour, ensuite, les mettre en perspective avec la grille d'observation et notre journal de bord.

5.2.1. Résultats de tous les élèves aux pré-test, post-test immédiat et post-test différé

Tableau 3 / Résultats de tous les élèves aux tests

Élève	Pré-test	Post-test immédiat			Post-test différé				
	Score obtenu	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Gain absolu	Perte / Gain relatifs (%)
				Par rapport au pré-test			Par rapport au pré-test		Par rapport au post-test immédiat
E1	15	23	8	100%	23	8	100%	0	0%
E2	21	18	-3	-14,29%	22	1	50%	4	80%
E3	16	20	4	57,14%	22	6	85,71%	2	66,67%
E4	21	23	2	100%	23	2	100%	0	0%
E5	8	21	13	86,67%	21	13	86,67%	0	0%
E6	23	23	0	0%	20	-3	-13,04%	-3	-13,04%
E7	19	21	2	50%	20	1	25%	-1	-4,76%
E8	12	19	7	63,64%	20	8	72,73%	1	25%
E9	20	22	2	66,67%	20	0	0%	-2	-9,09%
E10	21	23	2	100%	23	2	100%	0	0%
E11	17	16	-1	-5,88%	23	6	100%	7	100%
E12	21	23	2	100%	21	0	0%	-2	-8,70%
E13	14	19	5	55,56%	21	7	77,78%	2	50%
E14	23	22	-1	-4,35%	21	-2	-8,70%	-1	-4,55%
E15	18	23	5	100%	22	4	80%	-1	-4,35%
Moyenne	17,93	21,07	3,13*	57,01%	21,47	3,53	57,08%	0,40	18,48%

Dans cette recherche, nous avons tout d'abord émis l'hypothèse que le fait de proposer un scénario pédagogique composé d'activités débranchées et branchées auprès d'élèves du premier cycle de l'enseignement primaire serait un moyen efficace de développer leurs compétences en PIA.

Nous pouvons observer que 11 élèves, soit 73%, ont amélioré leur performance entre les deux tests, 3 élèves (E2, E11 et E14), soit 20%, ont régressé et un seul (E6), soit 7%, d'entre eux est resté au même score. Notons que ce dernier avait déjà un score maximum, score qu'il a conservé au post-test immédiat. Les élèves pour lesquels nous avons observé une baisse des résultats ont

quant à eux perdu un point pour deux d'entre-eux (E11 et E14) et trois points pour le dernier (E2). Nous reviendrons ultérieurement sur ces trois élèves dans le chapitre 5.2.5.

Si nous nous intéressons à la moyenne des résultats de tous les élèves, nous constatons une moyenne de 17,93/23 au pré-test et une moyenne de 21,07/23 au post-test immédiat, soit une augmentation de 3,14 points statistiquement significative ($p = 0,010$).

Par l'intermédiaire de ces premières données chiffrées, nous avons pu rendre compte de l'évolution globale de la réussite des apprenants. Pour chacune des compétences visées, la moyenne des résultats au terme de la mise en place de notre scénario pédagogique est plus élevée qu'avant son implémentation. En première analyse, nous pourrions dire que les élèves ont amélioré leurs compétences quant aux objectifs qui étaient visés par les apprentissages.

Néanmoins, nous avons affiné notre analyse des effets d'apprentissage en calculant des gains relatifs nous permettant de pouvoir comparer les résultats d'apprenants n'étant pas partis du même niveau de pré requis.

5.2.2. Analyse de l'effet d'apprentissage

Pour ce faire nous avons calculé un gain possible moyen ainsi qu'un gain brut moyen correspondant à ce qui a été effectivement gagné. Ces deux indices nous ont permis de calculer le gain relatif moyen exprimé en pourcentage et correspondant au rapport entre ce qui a été gagné et ce qui pouvait l'être.

Celui-ci s'élève à 57,01% pour l'ensemble des élèves. Tel que généralement admis dans la littérature scientifique, nous pouvons conclure à un effet positif sur l'apprentissage lorsque le gain relatif est supérieur à 30-35%. Ceci confirme l'hypothèse selon laquelle l'implémentation de notre scénario pédagogique a eu un effet positif sur le développement des compétences ciblées auprès des élèves de notre échantillon.

5.2.3. Analyse du coefficient de variation

Enfin, nous nous sommes intéressées au coefficient de variation permettant de rendre compte de la dispersion des résultats autour de la moyenne. Celui-ci correspond au rapport entre l'écart-type et la moyenne et permet de spécifier si les différences de scores entre les apprenants avant l'implémentation de notre scénario pédagogique se sont réduites au terme de celle-ci.

Le tableau ci-dessous (tableau 4) nous montre que le coefficient de variation passe de 23,09% à 10,16%, ce qui signifie que l'apprentissage a eu un effet d'équité au sein de notre échantillon puisque la variation interindividuelle se réduit.

Tableau 4 / Coefficient de variation des pré-test et post-test immédiat

N=15	Moyenne	Écart-type	Coefficient de variation
Pré-test	17,93	4,14	23,09 %
Post-test immédiat	21,07	2,14	10,16 %

5.2.4. La grille d'observation

Suite au codage des enregistrements vidéo (cf. annexe 7, p. 331), nous avons repris dans le tableau ci-dessous (tableau 5) le pourcentage des conduites observées pour chaque élève au regard de chaque compétence et sous-compétence travaillée au cours du scénario pédagogique. Ceci permet de rendre compte du poids de chacune d'entre-elles. De plus, le tableau 6 nous indique les interventions de l'enseignante, qu'elles soient verbales ou directement en lien avec le robot pédagogique.

Tableau 5 / Pourcentage des conduites observées pour chaque élève au regard de chaque compétence et sous-compétence travaillée

			E1	E2	E3	E4	E5	E6	E7	
Compétences	C.1	Sous-compétences	1.5	10%	6,41%	1,59%	7,06%	17,31%	12,50%	5,48%
	C.2		2.1	0%	0%	0%	0%	0%	0%	0%
			2.2	8%	6,41%	1,59%	5,88%	7,69%	0%	5,48%
			2.3	12%	26,92%	15,87%	16,47%	11,54%	16,67%	8,22%
			2.4	4%	11,54%	7,94%	11,76%	7,69%	12,50%	8,22%
	C.3		3.1	6%	12,82%	9,52%	8,24%	11,54%	12,50%	9,59%
	C.6		6.1	32%	17,95%	22,22%	16,47%	25%	37,50%	34,25%
			6.2	16%	7,69%	22,22%	17,65%	9,62%	8,33%	17,81%
			6.3	12%	10,26%	19,05%	16,47%	9,62%	0%	10,96%
			Total	100%	100%	100%	100%	100%	100%	100%
	N	50	78	63	85	52	24	73		

		E8	E9	E10	E11	E12	E13	E14	E15
Sous-compétences	1.5	4,94%	7,06%	3,53%	5,56%	6,74%	7,89%	9,26%	8,33%
	2.1	0%	0%	0%	0%	0%	0%	0%	1,39%
	2.2	2,47%	3,53%	0%	1,85%	4,49%	2,63%	1,85%	4,17%
	2.3	11,11%	11,76%	10,59%	20,37%	6,74%	18,42%	12,96%	19,44%
	2.4	7,41%	5,88%	11,76%	20,37%	11,24%	15,79%	5,56%	8,33%
	3.1	9,88%	8,24%	14,12%	11,11%	12,36%	18,42%	16,67%	12,50%
	6.1	33,33%	32,94%	30,59%	18,52%	23,60%	13,16%	24,07%	27,78%
	6.2	16,05%	20%	16,47%	12,96%	16,85%	7,89%	18,52%	11,11%
	6.3	14,81%	10,59%	12,94%	9,26%	17,98%	15,79%	11,11%	6,94%
	Total	100%	100%	100%	100%	100%	100%	100%	100%
N	81	85	85	54	89	38	54	72	

Tableau 6 / Pourcentages des interventions de l'enseignante (verbales ou directement en lien avec le robot pédagogique) pour chaque compétence et sous-compétence travaillée

				Facilitateur /	N	Facilitateur / Autres	N
				Sollicite			
Compétences	C.1	Sous-compétences	1.5	2,56%	3	1,14%	1
	C.2		2.2	1,71%	2	1,14%	1
			2.3	5,98%	7	7,95%	7
			2.4	11,11%	13	11,36%	10
	C.3		3.1	23,08%	27	18,18%	16
	C.6		6.1	29,06%	34	46,59%	41
			6.2	20,51%	24	4,55%	4
			6.3	5,98%	7	9,09%	8
			Total	100%	117	100%	88

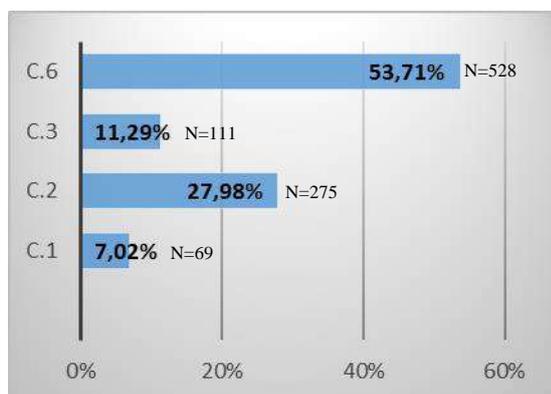


Figure 32 / Pourcentage des conduites observées liées à chaque compétence cible. N=983

Le graphique ci-contre (fig.32) nous indique que la fréquence des manifestations des conduites témoins associées à la compétence C.1 *Définir des abstractions / généraliser* représente 7,02% des conduites observées. Celles associées à la compétence C.2 *Composer/décomposer une séquence d'actions* représente 27,98% des actions analysées. Quant à celles associées à la compétence C.3 *Contrôler une séquence d'actions*, elle représente 11,29%. Enfin, celles associées à la

compétence C.6 *Construire une séquence d'actions de manière itérative* représente 53,71% des conduites observées, soit, un peu plus de la moitié.

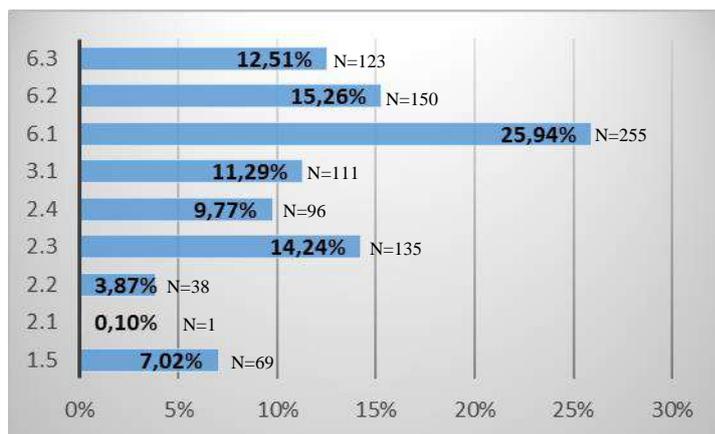


Figure 33 / Détail du pourcentage des conduites observées liées à chaque sous-compétence. N=983

Par ailleurs, comme l'indique le détail du pourcentage des conduites témoins observées pour chaque sous-compétence travaillée tout au long du scénario pédagogique (fig. 33), les élèves ont manifesté des conduites témoins associées aux sous-compétences visées.

La sous-compétence 6.1. *Vérifier si une séquence d'actions atteint un objectif donné* représente en moyenne un quart des conduites analysées. Cette dernière était visée lors de chaque séance. Rappelons que les items censés la mesurer avaient obtenu 73% de réussite au pré-test contre 94% et 97% aux post-test immédiat et différé, soit une augmentation de 21% et 24% suite à l'implémentation de notre dispositif.

S'en suivent les sous-compétences 2.3. *Créer une séquence d'actions pour atteindre un objectif simple*, 6.2. *Repérer des erreurs dans une séquence d'actions* et 6.3. *Corriger une séquence d'actions pour atteindre un objectif donné* qui représentent chacune en moyenne environ 15% des conduites analysées.

Les sous-compétences 2.4. *Créer une séquence d'actions pour atteindre un objectif complexe* fait référence au concept de boucle au même titre que la compétence 3.1. *Répéter une séquence d'actions un nombre donné de fois*. Travaillées uniquement les trois dernières séances, ceci peut expliquer qu'elles ne représentent chacune qu'environ 10% des conduites analysées.

Les sous-compétences 2.1. *Ordonner une séquence d'actions pour atteindre un objectif* et 2.2. *Compléter une séquence d'actions pour atteindre un objectif simple* ne faisaient pas partie des sous-compétences ciblées par notre scénario, ce qui explique leur faible pourcentage d'occurrences (0,10% et 3,87%).

En ce qui concerne la sous-compétence 1.5 *Prédire le résultat d'une séquence d'actions*, celle-ci n'est, quant à elle, travaillée qu'au cours de la dernière séance.

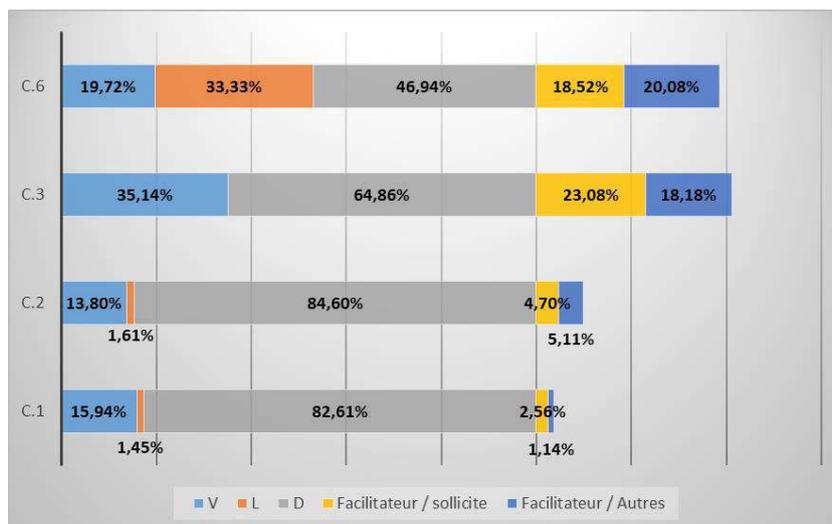


Figure 34 / Pourcentage des conduites observées en fonction des comportements pour chaque compétence & sollicitations de l'enseignante

Enfin, nous avons distingué les pourcentages des conduites observées en fonction des comportements verbaux (V), des actions sur le logiciel (L) ou sur le matériel didactique (D), et ce, pour chaque compétence visée. Le graphique ci-contre (fig. 34) nous indique que, pour chaque

compétence, les actions sur le matériel didactique représentent la majorité des comportements observés avec 82,61% pour la compétence C.1 *Définir des abstractions / généraliser*, 84,60% pour la compétence C.2 *Composer/décomposer une séquence d'actions*, 64,86% pour la compétence C.3 *Contrôler une séquence d'actions* et 46,94% pour la compétence C.6 *Construire une séquence d'actions de manière itérative*. Le pourcentage relatif des comportements verbaux est le plus important pour la compétence C.3 (23,08%) qui fait appel au concept de boucle. Ensuite, les actions sur le « logiciel », en l'occurrence notre robot, représentent un tiers (33,33%) de la compétence C.6. Ce résultat n'est pas surprenant puisque cette dernière amène les apprenants à tester leurs séquences.

Enfin, si nous nous penchons sur les conduites de l'enseignante, nous remarquons qu'elle émet très peu de sollicitations liées aux compétences C.1 et C.2. L'accent a été mis sur les deux autres compétences avec une part similaire entre les sollicitations verbales ou sur le matériel.

Pour conclure, et par l'intermédiaire de ces nouvelles données, nous pouvons déduire que l'implémentation de notre scénario pédagogique a eu pour effet la manifestation de conduites témoins de l'acquisition des compétences et sous-compétences visées.

5.2.5. Analyse des erreurs commises par les élèves présentant une perte relative entre le pré-test et le post-test immédiat.

Trois élèves de notre échantillon présentent une perte relative entre le pré-test et le post-test immédiat. Il s'agit des élèves E2 et E11 scolarisés en première année de l'enseignement primaire et tous deux de sexe masculin et d'E14, scolarisé en deuxième année et également de

sexe masculin. Notons que les élèves E2 et E11 faisait partie du même groupe de travail (P1B1) et que tous trois ont bénéficié de la seconde technique d'étayage consistant à écrire les commandes sur une feuille.

5.2.5.1. Élève n°2

Tableau 7 / Scores obtenus par E2 aux différents tests

	Pré-test	Post-test immédiat	Post-test différé
Réponses correctes	21	18	22
Réponses incorrectes	1	4	1
Abstentions	1	1	0

Tableau 8 / Items échoués ou abstenus par E2 aux différents tests

	Pré-test	Post-test immédiat	Post-test différé
Items échoués	9	20-21-22-23	21
Items abstenus	15	18	/

a) Analyse de l'item échoué au pré-test

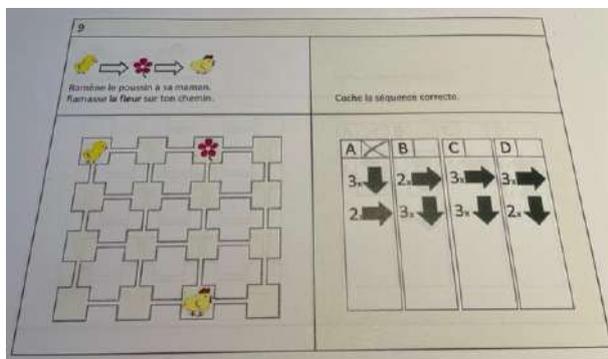


Figure 35 / Réponse d'E2 à l'item n°9 au pré-test

L'item 9 (fig.35) mesure les compétences 2.4. *Créer une séquence d'actions pour atteindre un objectif complexe*, 3.1. *Répéter une séquence d'actions un nombre donné de fois* et 6.1. *Vérifier si une séquence d'actions atteint un objectif donné*. Il combine un grand labyrinthe et un ramassage et représente l'item le plus compliqué pour mesurer le

concept de boucle. Son échec pourrait s'expliquer par le fait que ces notions n'aient pas encore été enseignées lors de la passation du pré-test.

b) Analyse des items échoués au post-test immédiat

Par ailleurs, les items 20 à 23 (cf. fig. 36 à 39) sont les quatre items mesurant la compétence 2.3. *Créer une séquence d'actions pour atteindre un objectif simple.*

Ceux-ci ont tous été réussis lors du pré-test, l'échec au post-test immédiat a de quoi nous interpeller.

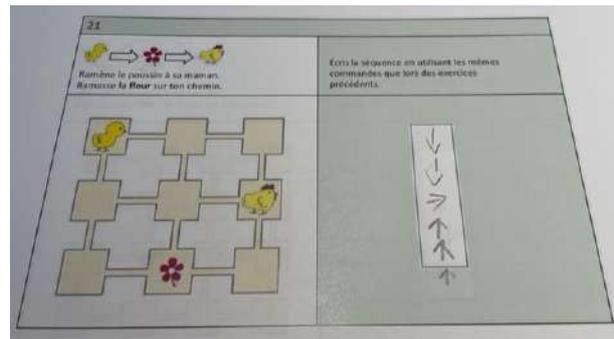


Figure 36 / Réponse d'E2 à l'item n°21 au post-test immédiat

Pourtant cet élève fait partie de ceux qui ont bénéficié de l'étayage consistant à écrire les commandes sur une feuille. Nous pourrions donc penser de prime abord que cela lui aurait été bénéfique dans le cadre de l'évaluation de cette compétence, hors, nous observons le contraire. Nous pourrions émettre plusieurs hypothèses, premièrement, que cet élève se soit retrouvé en difficulté puisqu'il se retrouve à écrire une séquence au départ d'un labyrinthe en deux dimensions alors qu'il s'est entraîné sur un labyrinthe en trois dimensions. Deuxièmement, que l'usage des flèches de rotation ait pu être un élément perturbateur dans la construction de son algorithme lors de l'évaluation.

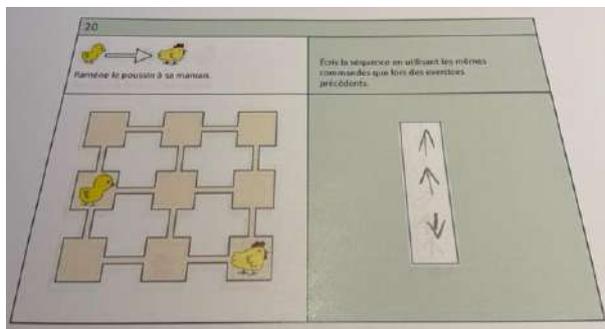


Figure 37 / Réponse d'E2 à l'item n°20 au post-test immédiat

Il semblerait que ce soit le cas si nous observons la réponse fournie à l'item 20 (fig.37). Nous pensons que l'élève a pris comme point de repère le nez du poussin pour le faire avancer deux fois et ensuite descendre pour atteindre l'objectif.

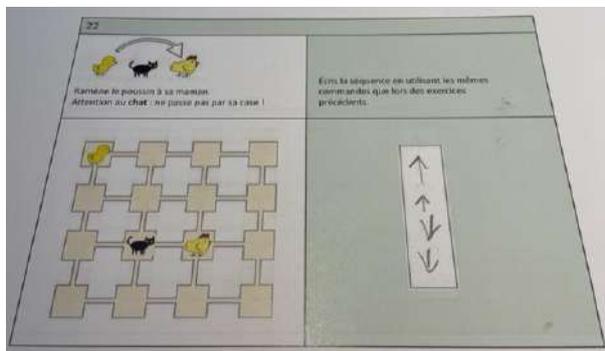


Figure 38 / Réponse d'E2 à l'item n°22 au post-test immédiat

La même technique semble avoir été répétée pour l'item 22 (fig.38) et l'item 23 (fig.39) avec cette nuance que pour le dernier, l'élève semble mélanger les commandes.

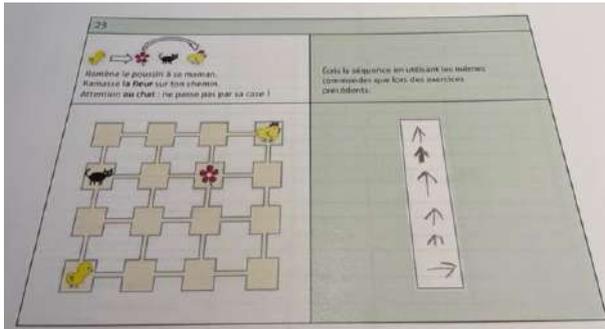


Figure 39 / Réponse d'E2 à l'item n°23 au post-test immédiat

En effet, nous pensons qu'il a commencé son algorithme par « avance, avance » en prenant comme point de repère le nez du poussin tel qu'il l'a fait lors des séances que nous avons dispensées. Ensuite, il semblerait qu'il change de code en inscrivant trois flèches vers le haut puis une flèche vers la droite pour le faire se déplacer comme cela était prévu.

Néanmoins, il est à noter que ce même élève a réussi les items 20, 22 et 23 lors du post-test différé.

c) Analyse de l'item échoué au post-test différé

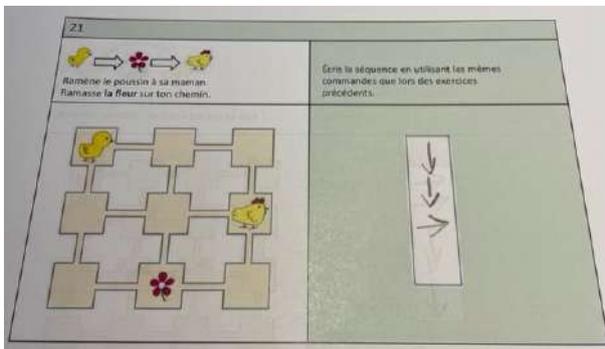


Figure 40 / Réponse d'E2 à l'item n°21 au post-test différé

Seul l'item 21 (fig.40) a, à nouveau, posé problème. Nous pouvons tout de même remarquer que l'algorithme construit était correct jusqu'à la fleur où il s'est arrêté.

d) Analyse des abstentions

Penchons-nous à présent sur les items auxquels l'élève s'est abstenu de répondre aux différents tests. Il s'agit de l'item 15 au pré-test et de l'item 18 au post-test immédiat.

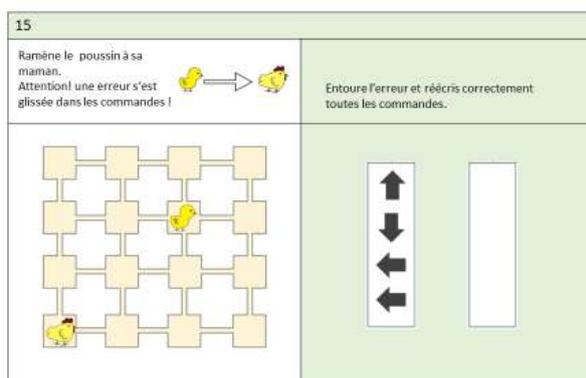
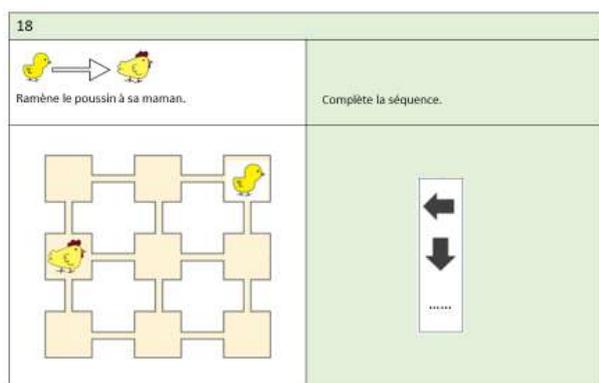


Figure 41 / Item n°15

L'item 15 (fig.41) qui évalue la compétence 6.2 *Vérifier si une séquence d'actions atteint un objectif donné* et 6.3 *Repérer des erreurs dans une séquence d'actions* est celui qui a été le moins bien réussi par tous les élèves lors du pré-test et dont nous avons déjà évoqué la problématique dans la partie 5.1.



L'item 18 (fig.42) évalue quant à lui la compétence 2.2 *Compléter une séquence d'actions pour atteindre un objectif simple* de même que l'item 19, plus compliqué, qui, lui, a été réussi.

Figure 42 / Item n°18

e) Liens entre les conduites témoins et les compétences

Nous allons analyser à présent le pourcentage des manifestations des conduites témoins associées aux compétences qui semblent avoir posé problème à cet élève, c'est-à-dire les sous-compétences 2.2 et 2.3.

La sous-compétence 2.2 représente 6,41% des conduites analysées chez E2, tandis que la sous-compétence 2.3 en représente 26,92%. Le potentiel manque de travail au niveau de cette dernière ne semble pas pouvoir expliquer l'échec aux items qui lui sont liés.

Cependant, comme la sous-compétence 2.2 n'a pas été travaillée spécifiquement au cours du dispositif, ceci pourrait expliquer que l'élève ait échoué à l'item 18. Néanmoins, si nous nous référons au schéma créé à partir de la matrice de Morganov proposant un ordre nous permettant d'aborder les compétences du référentiel PIAF (cf. annexe 8, p. 332), nous pourrions conclure que cette compétence soit considérée comme étant acquise lorsque les élèves ont acquis la sous-compétence 2.3.

5.2.5.2. Élève n°11

Tableau 9 / Scores obtenus par E11 aux différents tests

	<i>Pré-test</i>	<i>Post-test immédiat</i>	<i>Post-test différé</i>
<i>Réponses correctes</i>	17	16	23
<i>Réponses incorrectes</i>	5	7	/
<i>Abstentions</i>	1	/	/

Tableau 10 / Items échoués ou abstenus par E11 aux différents tests

	Pré-test	Post-test immédiat	Post-test différé
Items échoués	3-5-6-13-15	7-15-16-20-21-22-23	/
Items abstenus	7	/	/

a) Analyse des items échoués

L’item 15 semble également poser problème. Cet élève y échoue tant lors du pré-test et du post-test immédiat. L’ayant déjà analysé plus haut, nous ne reviendrons pas sur cet item.

Par contre, d’emblée, les items 20 à 23 (fig. 43 à 46) attirent notre attention. En effet, alors que E11 les a tous réussis lors du pré-test, l’élève échoue aux quatre items mesurant la compétence 2.3 *Créer une séquence d’actions pour atteindre un objectif simple* lors du post-test immédiat, et ce, au même titre qu’E2 qui faisait équipe avec lui. Nous allons analyser ses réponses de plus près.

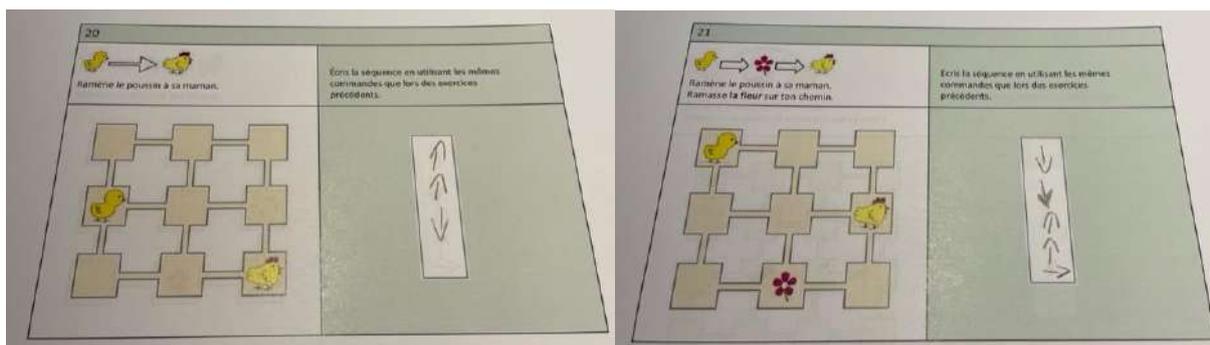


Figure 43 / Réponse d'E11 à l'item n°20 au post-test immédiat Figure 44 / Réponse d'E11 à l'item n°21 au post-test immédiat

Nous constatons que la réponse à l’item 20 (fig.43) est identique à celle rédigée par E2. Précisons qu’il leur était impossible de copier l’un sur l’autre au vu de la disposition des élèves lors de la passation des tests. Nous pourrions à nouveau émettre l’hypothèse que l’élève s’est également repéré au nez du poussin.



Figure 45 / Réponse d'E11 à l'item n°22 au post-test immédiat

Cette hypothèse pourrait se confirmer avec l’item 22 (fig.45) pour lequel l’élève a introduit « avance-avance-tourne à droite-avance-avance », algorithme qui aurait été correct lors de l’utilisation de la souris-robot.

Tableau 12 / Items échoués ou abstenus par E14 aux différents tests

	Pré-test	Post-test immédiat	Post-test différé
Items échoués	/	9	15-23
Items abstenus	/	/	/

a) Analyse des items échoués aux post-tests immédiat et différé

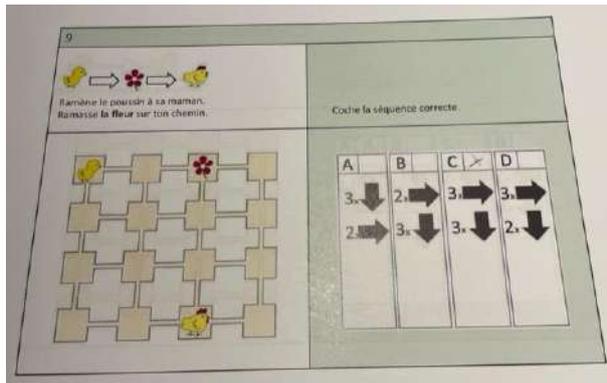


Figure 48 / Réponse d'E14 à l'item n°9 au post-test immédiat

L'analyse des trois items échoués chez cet élève (fig.48-49-50) permet de constater que la destination atteinte est systématiquement à côté du point final. L'élève utilise toujours une commande en trop dans la construction de son algorithme.

Doit-on y voir une simple erreur d'attention ?

À nouveau, nous pouvons constater que l'item 15 n'a pas été réussi.

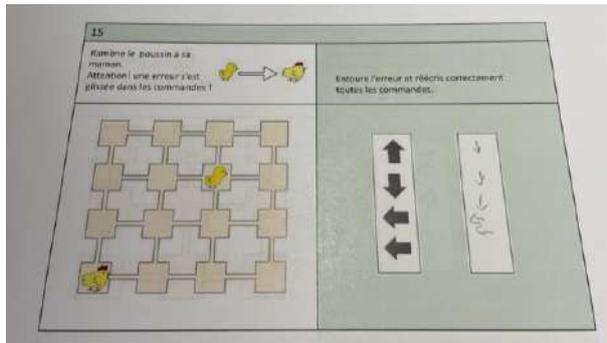


Figure 49 / Réponse d'E14 à l'item n°15 au post-test différé

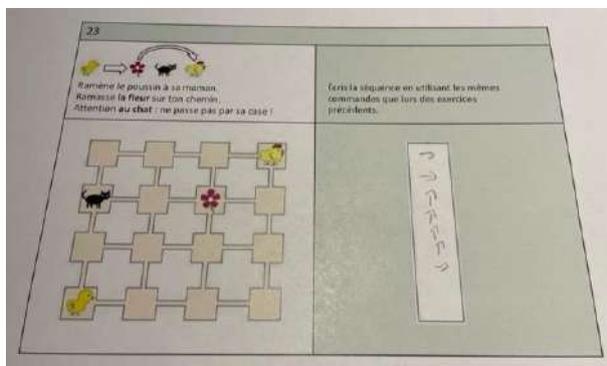


Figure 50 / Réponse d'E14 à l'item n°23 au post-test différé

b) Liens entre les conduites témoins et les compétences

Nous allons à nouveau analyser le pourcentage des comportements témoins de l'acquisition des compétences ciblées dans les items échoués par l'élève (fig.51)

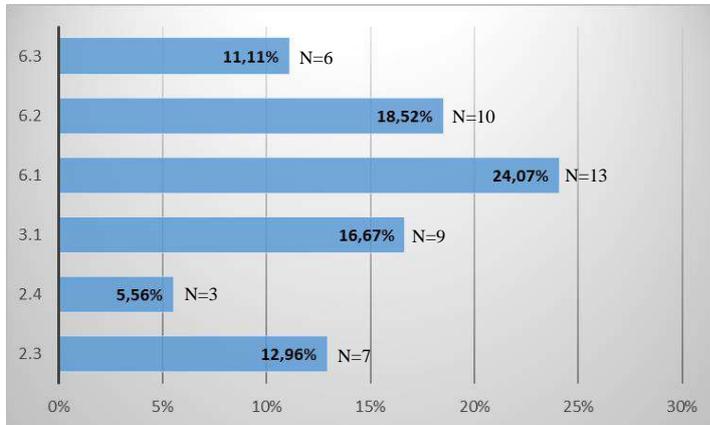


Figure 51 / Pourcentages des conduites témoins de l'acquisition des sous-compétences / E14. N=54

Le faible taux d'occurrence des conduites témoins de l'acquisition de la sous-compétence 2.4 nous amène à penser que cette dernière n'aurait pas assez été travaillée au cours du dispositif et pourrait expliquer le fait que cet élève ait échoué à l'item n°9. Rappelons qu'en moyenne cette sous-compétence représente 9,77% du total des conduites analysées et est

travaillée au cours des trois dernières leçons. Nous serions en droit de nous demander si le nombre de séances consacrées au travail de chaque compétence était suffisant pour permettre aux élèves de les acquérir.

Cependant, si nous nous intéressons à la sous-compétence 2.3 évaluée dans l'item 23 et travaillée tout au long de notre scénario pédagogique, nous pouvons constater une fréquence relative de 12,96%. Celle-ci a-t-elle été trop peu travaillée par cet élève ?

Notons cependant qu'il n'a pas assisté à la cinquième leçon au cours de laquelle les sous-compétences 2.3, 2.4, 3.1, 6.1, 6.2 et 6.3 étaient ciblées. Ces dernières sont toutes celles qui étaient mesurées par les items échoués par E14.

5.3. Maintien des compétences acquises dans le temps (H2)

Dans cette recherche, nous avons également émis l'hypothèse que les compétences acquises suite à l'implémentation de notre scénario pédagogique se maintenaient dans le temps. Pour ce faire, nous avons reproposé aux élèves un post-test différé un mois après le post-test immédiat.

Suite aux résultats obtenus à la suite des post-test immédiat et post-test différé des quinze élèves de notre échantillon (cf. tableau 3 p.60), nous pouvons observer que 3 élèves (E1, E4 et E10), soit 20%, ont conservé un score maximum au post-test différé. Un élève (E5), soit 7%, est resté au même résultat de 21 sur 23. Cinq élèves (E2, E3, E8, E11 et E13), soit 33%, ont amélioré leurs performances et 6 élèves, soit 40%, ont régressé au post-test différé. Trois d'entre eux

(E7, E14 et E15) ont échoué à un item, deux élèves (E9 et E12) à deux items et un élève (E6) a échoué à 3 items.

Si nous nous intéressons à la moyenne des résultats de tous les élèves, nous constatons une moyenne de 21,07/23 au post-test immédiat ainsi qu'une moyenne de 21,47 au post-test différé, soit une augmentation de 0,40 points entre le post-test immédiat et le post-test différé ($p = 0,554$). En première analyse nous pourrions conclure qu'à la suite de l'implémentation de notre scénario pédagogique, les compétences visées acquises semblent se maintenir dans le temps.

Tel que nous l'avons fait pour la première hypothèse, nous avons également calculé les gains relatifs entre le pré-test et le post-test différé ainsi qu'entre le post-test immédiat et le post-test différé.

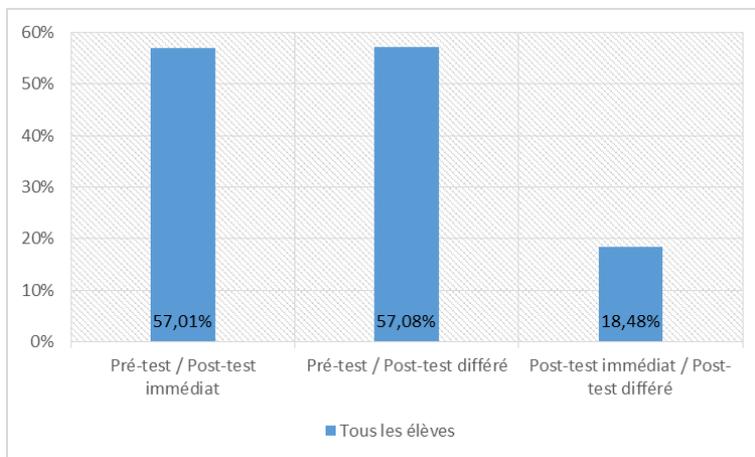


Figure 52 / Gains relatifs moyens pour tous les élèves

Dans la figure ci-dessus (fig.52), nous pouvons observer un gain relatif moyen de 57,01% entre le pré-test et le post-test immédiat. Celui-ci passe à 57,08% entre le pré-test et le post-test différé. La différence entre les deux post-tests s'élève à 18,48%.

Nous pouvons donc conclure que les compétences visées et acquises tendent à se maintenir dans le temps, voire à progresser puisque nous observons un gain relatif moyen plus élevé en comparaison au post-test immédiat.

5.4. Acquisition des compétences en PIA selon la technique d'étayage (H3)

5.4.1. Résultats des élèves ayant bénéficié de la première technique d'étayage : utilisation des cartes de programmation

Tableau 13 / Résultats des élèves en première condition d'étayage

Élève	Pré-test	Post-test immédiat			Post-test différé				
	Score obtenu	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Gain absolu	Perte / Gain relatifs (%)
E1	15	23	8	100%	23	8	100%	0	0%
E5	8	21	13	86,67%	21	13	86,67%	0	0%
E6	23	23	0	0%	20	-3	-13,04%	-3	-13,04%
E7	19	21	2	50%	20	1	25%	-1	-4,76%
E10	21	23	2	100%	23	2	100%	0	0%
E12	21	23	2	100%	21	0	0%	-2	-8,70%
E13	14	19	5	55,56%	21	7	77,78%	2	50%
E15	18	23	5	100%	22	4	80%	-1	-4,35%
Moyenne	17,38	22	4,63*	74,03%	21,38	4	57,05%	-0,63	2,39%

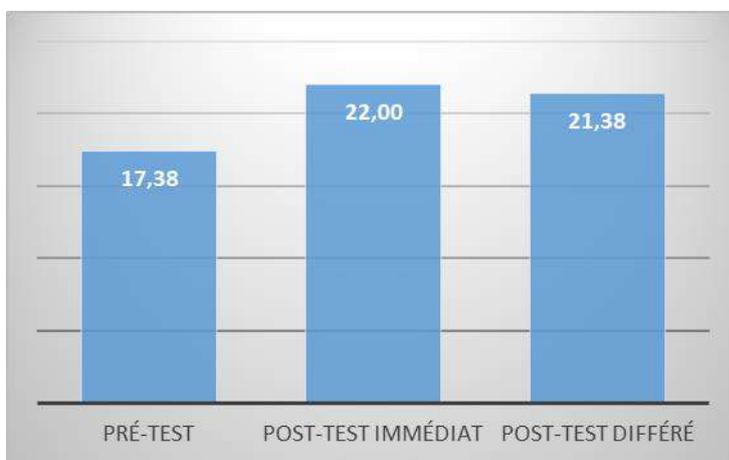


Figure 53 / Évolution des moyenne aux différents tests en première condition d'étayage

En observant ce graphique (fig.53), nous pouvons constater une augmentation de la moyenne des résultats entre le pré-test et le post-test immédiat avec une différence de 4,62 points, et ce, dans la première condition d'étayage. Le calcul du t de Student nous indique que cette différence est statistiquement significative ($p = 0,171$).

Cette moyenne passe à 21,38 pour le post-test différé, soit une perte de 0,62 points ($p = 0,279$).

5.4.2. Résultats des élèves ayant bénéficié de la seconde technique d'étayage : l'écriture

Tableau 14 / Résultats des élèves en seconde condition d'étayage

Élève	Pré-test	Post-test immédiat			Post-test différé				
	Score obtenu	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Score obtenu	Gain absolu	Perte / Gain relatifs (%)	Gain absolu	Perte / Gain relatifs (%)
E2	21	18	-3	-14,29%	22	1	50%	4	80%
E3	16	20	4	57,14%	22	6	85,71%	2	66,67%
E4	21	23	2	100%	23	2	100%	0	0%
E8	12	19	7	63,64%	20	8	72,73%	1	25%
E9	20	22	2	66,67%	20	0	0%	-2	-9,09%
E11	17	16	-1	-5,88%	23	6	100%	7	100%
E14	23	22	-1	-4,35%	21	-2	-8,70%	-1	-4,55%
Moyenne	18,57	20	1,43	37,56%	21,57	3	57,11%	1,57	36,86%

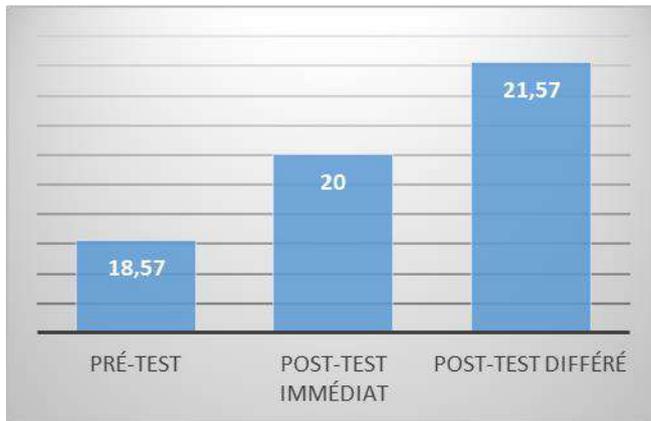


Figure 54 / Évolution des moyennes aux différents tests en seconde condition d'étayage

L'observation de ce graphique (fig.54) indique quant à lui également une augmentation de la moyenne des résultats entre le pré-test et le post-test immédiat avec une différence de 1,43 points, et ce, dans la seconde condition d'étayage ($p = 0,310$). Cette moyenne passe à 21,57 pour le post-test différé avec une nouvelle augmentation de 1,57 points par rapport au post-test immédiat ($p = 0,228$).

5.4.3. Analyse des effets d'apprentissage en fonction de la technique d'étayage reçue

Nous avons à nouveau affiné notre analyse des effets d'apprentissage en calculant des indices de gains entre les deux techniques d'étayage qui ont été utilisées.

Pour ce faire nous avons procédé de la même manière que pour les autres hypothèses.

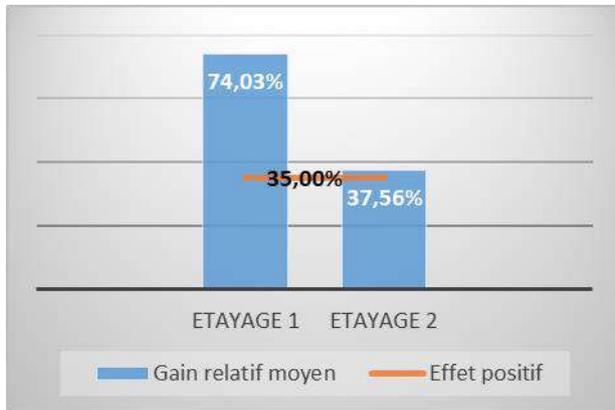


Figure 55 / Gain relatif moyen en fonction de la technique d'étayage

L'analyse du gain relatif moyen présent dans la figure ci-contre (fig.55) permet de mieux appréhender l'effet de l'apprentissage pour chacune des techniques d'étayage mises en place. On observe un gain de 74,03% pour la première technique d'étayage comparativement à un gain de 37,56% pour la seconde technique d'étayage. Cette différence de 36,47% plaide en faveur de la première technique d'étayage dans laquelle,

rappelons-le, les élèves ont eu l'occasion de manipuler des petites cartes de programmation. De plus, tel que généralement admis dans la littérature scientifique, nous pouvons conclure à un effet positif sur l'apprentissage lorsque le gain relatif est supérieur à 30-35%. Il semblerait donc, selon ces analyses, que la deuxième technique d'étayage, utilisant l'écriture, présente un effet tout juste positif.

5.4.4. Analyse des occurrences liées à la grille d'observation

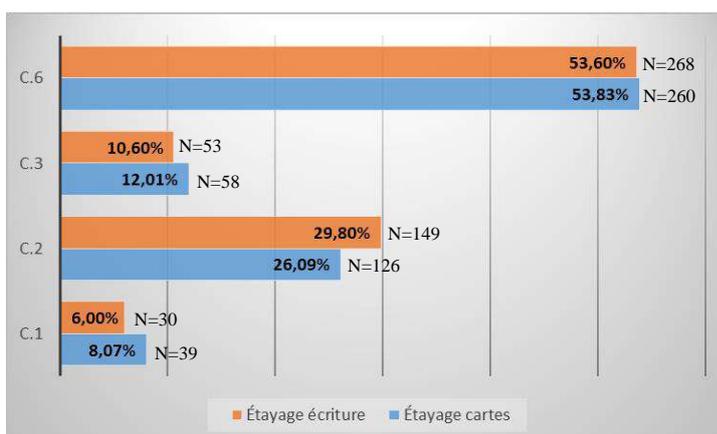


Figure 56 / Pourcentages des comportements observés pour chaque compétence en fonction de la technique d'étayage

Le graphique ci-contre (fig.56) ne nous indique pas d'écart significatif au niveau du pourcentage des comportements observés pour chaque compétence en fonction de la technique d'étayage dont les apprenants ont bénéficié. Il semblerait que celle-ci n'ait pas eu d'impact sur la répartition des conduites témoins de l'acquisition des compétences en PIA.

5.5. Acquisition des compétences en PIA en fonction de la technique d'étayage et du genre (H4)

Nous avons procédé au même type d'analyse en fonction de la technique d'étayage mais également du genre auquel appartiennent les élèves.

5.5.1. Résultats des filles dans les deux conditions d'étayage

Tableau 15 / Résultats des filles dans les deux conditions d'étayage

	Élèves ♀	Pré-test		Post-test immédiat	
		Score obtenu	Score obtenu	Gain absolu	Perte / Gain relatifs (%)
		Par rapport au pré-test			
Étayage n°1	E5	8	21	13	86,67%
	E7	19	21	2	50%
	E10	21	23	2	100%
	E13	14	19	5	55,56%
	E15	18	23	5	100%
	Moy.	16	21,4	5,4	78,45%
Étayage n°2	E3	16	20	4	57,14%
	E4	21	23	2	100%
	E8	12	19	7	63,64%
	Moy.	16,33	20,67	4,33	73,59%

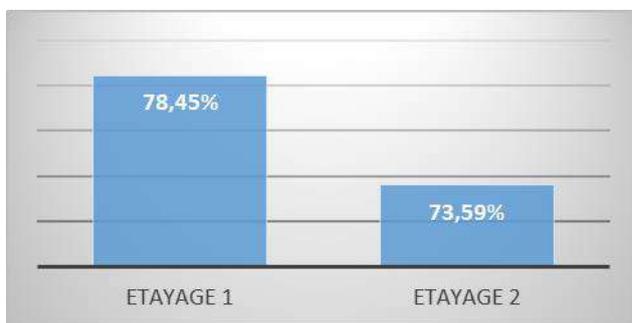


Figure 57 / Gains relatifs moyens des filles suivant l'étayage

Le graphique ci-contre (fig.57) nous indique au post-test immédiat un gain relatif moyen de 78,45% en première condition d'étayage contre 73,59% en seconde condition d'étayage. Soit une différence de 4,89%. Les deux techniques proposées semblent donc avoir un effet positif sur l'apprentissage avec un léger avantage pour la première.

la première.

5.5.2. Analyse des occurrences liées à la grille d'observation chez les filles

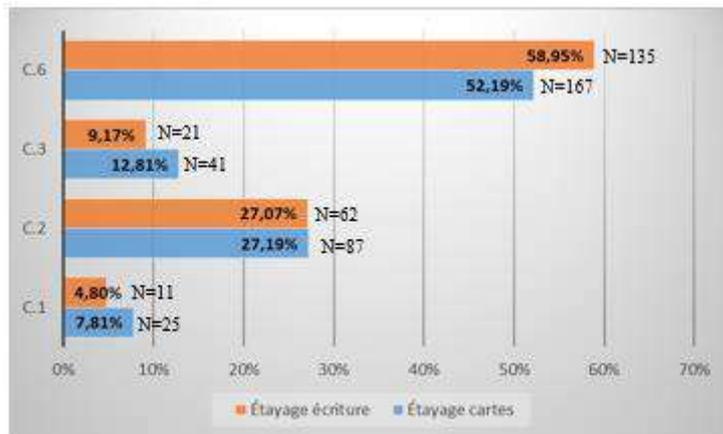


Figure 58 / Pourcentage des conduites observées liées à chaque compétence cible en fonction de la technique d'étayage chez les filles. Étayage écriture : N=229 ; Étayage cartes : N=320

Le graphique ci-contre (fig.58) rend compte du pourcentage des conduites observées liées à chaque compétence cible en fonction de la technique d'étayage. Nous ne pouvons pas constater de réelle différence en termes de pourcentages. Cependant nous pouvons noter une différence en termes d'occurrences puisque leur nombre est plus élevé en première

condition d'étayage qu'en deuxième avec respectivement 320 et 229 occurrences, soit une différence de 91.

5.5.3. Résultats des garçons dans les deux conditions d'étayage

Tableau 16 / Résultats des garçons dans les deux conditions d'étayage

	Élèves ♂	Pré-test		Post-test immédiat	
		Score obtenu	Score obtenu	Gain absolu	Perte / Gain relatifs (%)
Étayage n°1	E1	15	23	8	100%
	E6	23	23	0	0%
	E12	21	23	2	100%
	Moy.	19,67	23	3,33	66,67%
Étayage n°2	E2	21	18	-3	-14,29%
	E9	20	22	2	66,67%
	E11	17	16	-1	-5,88%
	E14	23	22	-1	-4,35%
	Moy.	20,25	19,50	-0,75	10,54%

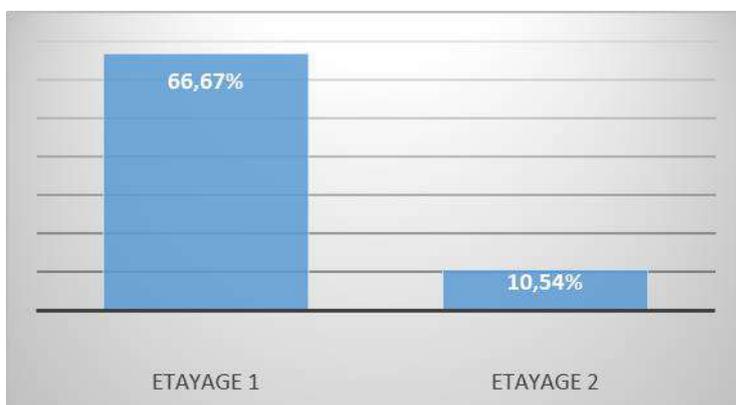


Figure 59 / Gains relatifs moyens des garçons en fonction de l'étayage

La figure ci-contre (fig.59) nous montre par ailleurs que le gain relatif moyen des garçons en fonction de la technique utilisée est nettement supérieur pour la première technique d'étayage avec un gain relatif moyen de 66,67% en

comparaison à la seconde qui, quant à elle, nous indique un gain relatif moyen de 10,54%, soit une différence de 56,13%.

Ces résultats sont toutefois à nuancer puisque trois élèves sur les quatre ayant bénéficié de la seconde technique d'étayage ont observé une perte relative, qui, suite à l'analyse que nous avons effectuée plus haut nous indique que les élèves ont tout de même acquis les compétences qui étaient visées, mais que les consignes des items de notre évaluation auraient pu avoir prêté à confusion.

5.5.4. Analyse des occurrences liées à la grille d'observation chez les garçons

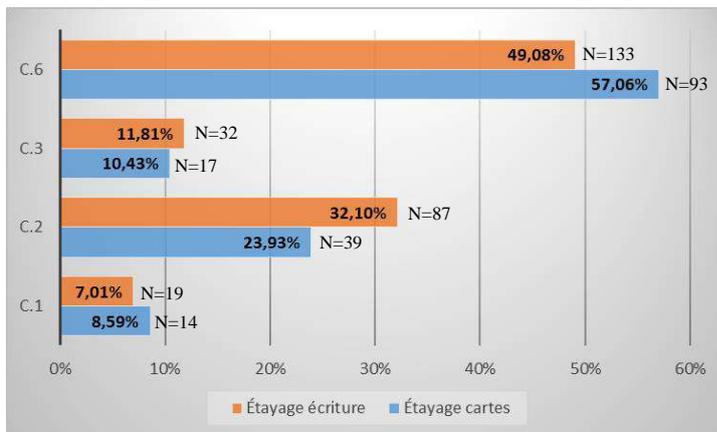


Figure 60 / Pourcentage des conduites observées liées à chaque compétence cible en fonction de la technique d'étayage chez les garçons. Étayage écriture : N=271 ; Étayage cartes : N=163

Le graphique ci-contre (fig.60) rend compte du pourcentage des conduites observées liées à chaque compétence cible en fonction de la technique d'étayage. À nouveau, tel que nous avons pu l'observer pour les filles, nous ne pouvons constater de réelle différence en termes de pourcentages. Néanmoins, nous voyons apparaître que le nombre d'occurrences en seconde condition

d'étayage est plus élevé qu'en première condition avec respectivement 271 contre 163, soit une différence de 108.

Ces résultats vont à l'encontre de ceux observés chez les filles. Il semblerait que ces dernières manifestent davantage de conduites témoins de l'acquisition des compétences cibles en première condition d'étayage (faisant usage des cartes), alors que les garçons en manifesteraient plus en seconde condition d'étayage (faisant usage de l'écriture).

6. Discussion

Suite à la présentation et à l'analyse de nos résultats, nous allons à présent les discuter afin de les mettre en perspective avec notre revue de littérature et les différentes hypothèses que nous avons émises.

6.1. L'évaluation

Suite à l'analyse de notre évaluation, nous avons relevé plusieurs points de questionnements qu'il nous semble important de mettre en lumière.

Premièrement, sur l'ensemble des résultats obtenus au pré-test par tous les élèves, nous observons un taux de réussite relativement élevé. Nous serions donc en droit de nous demander si le niveau de l'évaluation que nous avons conçue est adéquat pour les élèves de notre échantillon, celui-ci étant composé d'élèves scolarisés en première et en deuxième année primaire. Cependant, si nous prenons en considération la première partie reprise du Beginners Computational Thinking Test (BCTt) conçu par Román-González et al. (2020) et que nous nous référons à leurs propos, nous pouvons conclure que les items peuvent être utilisés auprès d'élèves de niveau primaire, particulièrement chez ceux âgés de 5 à 10 ans jusqu'aux élèves plus âgés (de 9 à 12 ans). Nous rejoignons cependant Poulakis & Politis (2021) en disant qu'il est nécessaire de multiplier les recherches en s'appuyant sur la littérature scientifique afin d'obtenir des outils validés en tenant compte du groupe d'âge pour lequel ils ont été conçus.

Deuxièmement, tels que Román-González et al. (2017) plaident en leur faveur, la multiplicité des méthodes d'évaluation apparaît comme étant l'approche la plus congruente pour évaluer les compétences en pensée informatique. C'est pourquoi nous avons pris le parti de les évaluer au travers d'un pré-test, un post-test immédiat ainsi qu'un post-test différé que nous avons construit en adéquation avec les méthodes et les objectifs poursuivis dans notre scénario pédagogique mais également à l'aide d'une grille d'observation présentant des critères observables permettant de juger de l'apparition de conduites témoins de leur acquisition. Enfin, nous avons tenu un journal de bord tout au long de notre recherche afin d'y référencer les remarques et points d'attention qui nous ont interpellés lors de chaque séance. Les différents points de vue que nous avons pu référencer dans les différentes méthodes d'évaluation nous ont permis d'affiner et de nuancer nos résultats et nos observations.

6.2. Acquisition des compétences en PIA (H1)

Dans cette recherche, nous avons tout d'abord fait l'hypothèse que les élèves progressent dans l'acquisition des compétences de pensée informatique et algorithmique suite à l'implémentation de scénarios pédagogiques spécifiques au développement de celles-ci.

Sur base de l'analyse de l'effet d'apprentissage couplée à la mise en évidence de conduites témoins de l'acquisition des compétences spécifiques en PIA à travers la grille d'observation, nous pouvons confirmer notre première hypothèse. Les données collectées nous indiquent que les compétences et sous-compétences ciblées par notre scénario pédagogique ont bel et bien été mobilisées par les élèves de notre échantillon. À l'instar des recherches menées par Angeli et Valadines (2020) qui, à la suite d'activités de robotique pédagogique, font état de gains statistiquement significatifs entre les deux évaluations mesurant les compétences en pensée informatique, nous avons également observé une différence statistiquement significative entre le pré-test et le post-test immédiat ($p = 0,010$).

Ceci rejoint le champ des études actuelles qui indiquent, d'une part, que les activités de robotique pédagogique constituent une approche intéressante pour développer la pensée informatique des jeunes enfants (Baeck et al., 2019 ; Beraza et al., 2010 ; Bers et al., 2014 ; Stoeckelmayer et al. 2011) et , d'autre part, que le fait de proposer aux élèves un enseignement basé sur des activités mixte s'avère bénéfique en termes d'acquisition des compétences en PIA (del Olmo-Muñoz et al., 2020).

De plus, si nous nous attardons sur l'analyse des erreurs commises par les élèves ayant obtenus une perte relative entre leur score au pré-test et au post-test immédiat, ceci nous amène à nous questionner sur l'utilisation du code qui a été mis en œuvre au regard de l'évaluation qui a été proposée. Celui-ci a-t-elle joué un rôle dans la moyenne des gains relatifs que nous avons observée ? Il semblerait en effet que les items 20 à 23 aient prêté à confusion pour certains élèves, ce qui a eu comme conséquence une influence de ces pertes négatives sur le calcul de l'effet de l'apprentissage.

Enfin, nous soulevons également la question de la durée de l'apprentissage permettant aux élèves d'acquérir les compétences en PIA.

6.3. Maintien des compétences acquises dans le temps (H2)

Ensuite, nous avons émis l'hypothèse que l'acquisition de ces compétences se maintient dans le temps. À la lumière des résultats que nous avons obtenus suite à la passation du post-test différé, nous nous devons de nuancer les conclusions de cette hypothèse. En effet, un tiers des élèves de notre échantillon observe une amélioration, un autre tiers se maintient, tandis que le dernier tiers observe une baisse des résultats, et ce, en comparaison avec les résultats qu'ils ont obtenus lors du post-test immédiat. Ceci soulève à nouveau la question de l'évaluation. Il conviendrait de pousser les investigations afin d'analyser plus en profondeur les items qui ont posé problème aux élèves lors de cette dernière évaluation.

Néanmoins, l'analyse du gain relatif nous indique une différence de 18,48% entre le post-test immédiat et le post-test différé, et ce, en faveur de ce dernier. Ceci permet de contribuer à atteindre l'un des objectifs de l'éducation qui est de pouvoir amener les apprenants à transférer les connaissances et compétences acquises en classe à d'autres matières ou d'autres contextes dans ou à l'extérieur de celle-ci grâce à un apprentissage qui se veut durable et qui se perdure après l'évaluation immédiate (Halpern & Hakel, 2002).

6.4. Acquisition des compétences en PIA selon la technique d'étayage (H3)

Aussi, nous avons émis l'hypothèse que l'acquisition des compétences spécifiques à la PIA diffère selon la technique d'étayage mise en œuvre. Pour ce faire, nous en avons développé deux en prenant en considération les caractéristiques techniques de notre robot ainsi que le contexte de notre implémentation. Pour rappel, la première technique d'étayage a été conçue de façon à fournir à chaque apprenant des cartes de programmation reprenant les quatre commandes directionnelles, tandis que la seconde technique d'étayage a été conçue afin de permettre aux apprenants d'écrire leur algorithme avant de pouvoir le tester.

À l'instar de la recherche menée par Angeli et Valadines (2020), nos résultats montrent que tous les élèves ont obtenus des gains d'apprentissage quelle que soit la technique d'étayage qui leur a été proposée.

Nos résultats sont néanmoins à mettre en perspective avec plusieurs aspects. Le premier aspect à observer serait le niveau de difficulté de la tâche qui leur était proposée. En effet, bien que les aides visuelles restent un moyen efficace de s'engager dans les différents niveaux de pensée informatique et tels que Chou et Shih (2021) l'ont avancé dans leur étude, le recours à celles-ci

ne serait efficace que lorsque les apprenants se verraient confrontés à des problèmes nécessitant une réflexion de niveau supérieur.

Le deuxième aspect à observer serait les conditions organisationnelles de notre recherche puisque tous les élèves ont été répartis en dyade ou en triade. Ceci aurait pu avoir comme effet d'influencer le développement et l'acquisition des compétences en pensée informatique comme souligné par plusieurs auteurs (Denner et al., 2014 ; Hanks, 2008 ; Wei et al., 2021). La répartition aléatoire des apprenants a-t-elle joué un rôle ? Les élèves auraient-ils progressé de la même façon s'ils avaient été répartis avec d'autres camarades, qu'ils soient ou non du même genre ou de la même année scolaire ?

Enfin, le troisième aspect à observer serait le niveau d'écriture de nos élèves. Le fait de proposer un étayage basé sur l'écriture à des élèves n'ayant pas encore automatisé le geste graphique engendre-t-il une charge cognitive supplémentaire ?

6.5. Acquisition des compétences en PIA en fonction de la technique d'étayage et du genre (H4)

Enfin, nous avons émis l'hypothèse que l'implémentation de ces deux techniques d'étayage va jouer un rôle différent sur le développement des compétences en PIA des élèves en fonction du genre.

Suite aux résultats que nous avons obtenus (cf. p.78), nous pouvons conclure que nous n'observons pas de différence de résultats entre les deux techniques d'étayage reçues par les filles. Cependant, si nous nous référons au calcul du gain d'apprentissage, les garçons ont, quant à eux, plus profité de la première technique d'étayage faisant usage des cartes de programmation. Ce constat rejoint en partie celui d'Angeli et Valadines (2020) puisqu'ils observent un effet d'interaction statistiquement significatif entre le genre et l'étayage dans lequel les garçons ont bénéficié davantage de la technique d'étayage basée sur la manipulation des cartes de programmation tandis que les filles ont bénéficié davantage de l'écriture collaborative.

Il convient toutefois de nuancer ces comparaisons puisque les deux techniques d'étayage n'étaient pas identiques dans les deux recherches. En effet, la seconde technique proposée par Angeli et Valadines (2020) était basée sur l'écriture collaborative où l'enseignant était en charge de noter ce que l'apprenant lui dictait tandis que dans notre recherche, l'apprenant devait

noter lui-même ses commandes. Il semblerait donc que la première technique d'étayage ait été plus propice à développer les compétences en PIA.

Lorsque nous nous intéressons aux conduites témoins de l'acquisition des compétences cible, nous observons que les filles en manifesteraient davantage en première condition d'étayage (faisant usage des cartes), alors que les garçons en manifesteraient plus en seconde condition d'étayage (faisant usage de l'écriture). C'est le même constat qui est fait par Papavlasopoulou et al., 2020 qui, rappelons-le, affirment que les filles ont simplement une approche et des stratégies différentes lors des activités de pensée informatique. Dès lors, nous pourrions rejoindre ces propos. Néanmoins, la question de l'évaluation se doit à nouveau d'être soulevée puisque, rappelons-le, bien qu'en partie validée scientifiquement, certains items ont été construits dans le cadre de cette recherche et nous avons pu constater que certains d'entre eux semblaient dysfonctionner. La seconde technique d'étayage aurait pu prêter à confusion dans la construction des algorithmes.

6.6. Perspectives

A l'issue de notre recherche, nous n'avons pas la prétention d'avoir approché tous les contours de l'enseignement et apprentissage de la pensée informatique et algorithmique. Néanmoins, notre travail donne lieu à plusieurs interrogations qui mériteraient d'être investiguées lors de recherches ultérieures.

Premièrement, comme nous avons eu l'occasion de le souligner à plusieurs reprises, il conviendrait d'investiguer plus en profondeur la question de l'évaluation de la pensée informatique et plus particulièrement de notre évaluation qui mériterait de faire l'objet d'une analyse plus approfondie pour vérifier sa validité et sa fiabilité.

Deuxièmement, nous soulevons la question de la durée des apprentissages dans l'acquisition des compétences en PIA.

Troisièmement, nous pensons qu'il pourrait être intéressant de différencier les résultats en termes d'année de scolarisation. Notre échantillon étant composé d'élèves de première et deuxième année de l'enseignement primaire, nous soulevons la question d'un âge qui serait plus propice à l'acquisition de telle ou telle compétence. Ceci permettrait également de s'interroger à nouveau sur une évaluation qui pourrait être plus adéquate en fonction du niveau des élèves mais aussi sur les techniques d'étayage qui pourraient être plus appropriées.

Quatrièmement, un autre domaine de recherche est lié à la nécessité de multiplier les techniques d'étayage afin de ne pas se limiter.

Cinquièmement, nous pensons également qu'il serait pertinent de mettre à profit l'analyse du journal de bord que nous avons construit tout au long de notre recherche et que nous n'avons malheureusement pas eu le temps d'exploiter pleinement. Ce dernier renferme des informations qui pourraient être utiles dans la compréhension des processus par lesquels les apprenants évoluent dans la construction du développement de leurs compétences en PIA, dans le niveau de difficulté de la tâche proposée ou encore du nombre de tentatives nécessaires à la réussite des exercices.

Enfin, il conviendrait de reproduire cette recherche dans un contexte plus vaste avec un échantillon plus grand.

7. Conclusion

Les compétences en lecture, en écriture ou en mathématiques sont indispensables depuis 200 ans. Cependant, dans un monde qui évolue en permanence, il est nécessaire de développer de nouvelles compétences du 21^e siècle comme la pensée critique, la créativité, la collaboration, la résolution de problèmes et la pensée informatique (Romero, 2017).

Afin d'explorer les effets de l'implémentation d'un scénario pédagogique couplé à deux techniques d'étayage différentes sur l'acquisition des compétences spécifiques liées à la pensée informatique et algorithmique, nous avons mené une recherche auprès de 15 apprenants scolarisés en première et deuxième année de l'enseignement primaire. Pour cette étude, les élèves ont participé à sept leçons composées d'activités débranchées et branchées selon deux modalités d'étayage différentes afin de vérifier si l'une ou l'autre avait un impact sur le développement des compétences en PIA.

Les résultats que nous avons pu mettre en lumière ont permis d'alimenter plusieurs domaines de recherches tels que présentés par Angeli & Giannakos (2020).

Premièrement, le domaine des stratégies pédagogiques. S'il est important d'administrer un étayage approprié aux élèves aux cours de leur apprentissage, l'un des enjeux réside dans le fait de multiplier les approches pédagogiques afin de répondre au mieux à la diversité du public auquel les enseignants doivent faire face sans pour autant tomber dans une espèce de recette toute faite qui conviendrait à tout un chacun.

Gardons néanmoins à l'esprit que les neuromythes peuvent avoir certaines répercussions regrettables dans le domaine de l'éducation en incitant par exemple les enseignants à porter leurs choix pédagogiques vers des méthodes moins efficaces ou encore en associant les apprenants à des classifications réductrices, voire même stigmatisantes, telles que les stéréotypes de sexe (Blanchette Sarrasin & Masson, 2017).

Deuxièmement celui du développement professionnel des enseignants. La création et la dissémination de scénarios pédagogiques spécifiques auprès des enseignants peut être un moyen efficace de développer les compétences en PIA de leurs élèves. Ceci pourrait permettre également de démystifier le concept de PIA auprès des enseignants qui se verraient outillés pour être à même de pouvoir mettre en place des activités liées à la pensée informatique dans leur classe.

Enfin, celui de l'évaluation de la pensée informatique qui reste un défi majeur à relever.

En conclusion, les besoins du monde de demain ne sont plus les mêmes que ceux d'hier. Pour en affronter les enjeux, la pensée informatique semble être une clé efficace. Espérons que la communauté enseignante, en s'appuyant sur le Pacte pour un Enseignement d'excellence, en perçoive les enjeux et fournisse aux élèves les clés pour devenir les citoyens actifs de demain.

8. Bibliographie

- Amadiou, F., & Tricot, A. (2020). *Apprendre avec le numérique : Mythes et réalité (2e éd.)*. Éditions Retz. DOI: [10.14375/NP.9782725638768](https://doi.org/10.14375/NP.9782725638768)
- Angeli, C., & Giannakos, M. (2020). Computational thinking education: Issues and challenges. *Computers in Human Behavior, 105*, 106185.
- Angeli, C., & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in human behavior, 105*, 105954.
- Atmatzidou, S., & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems, 75*, 661-670.
- Baek, Y., Wang, S., Yang, D., Ching, Y. H., Swanson, S., & Chittoori, B. (2019). Revisiting second graders' robotics with an understand/use-modify-create (U2MC) strategy. *European Journal of STEM Education, 4*(1).
- Beaugrand, J. (1988). Observation directe du comportement. In M. Robert (dir.), *Fondements et étapes de la recherche scientifique en psychologie (3' édition)* (p. 277-310). Saint-Hyacinthe: Edisem. (1^{re} éd. 1982)
- Beraza, I., Pina, A., & Demo, B. (2010). Soft & Hard ideas to improve interaction with robots for Kids & Teachers. *Proceedings of SIMPAR, 549-557*.
- Berger, K. S. (2010). *Psychologie du développement (2e éd.)*. Montréal, Québec : Modulo éditeur.

- Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.
- Benbow, C. P. & Lubinski, D. S., (2007). Sex differences in personal attributes for the development of scientific expertise. In S. J. Ceci & W. M. Williams (Eds.), *Why aren't more women in science? Top researchers debate the evidence* (pp. 79–100). Washington, DC: American Psychological Association. doi: 10.1037/11546-007
- Blanchette Sarrasin, J., & Masson, S. (2017). Connaître les neuromythes pour mieux enseigner. *Enjeux pédagogiques*, 28, 16-18.
- Bonomo, V. (2010). Gender matters in elementary education research-based strategies to meet the distinctive learning needs of boys and girls. *Educational Horizons*, 88(4), 257-264.
- Broisin, J., Declercq, C., Fluckiger, C., Parmentier, Y., Peter, Y., & Secq, Y. (2021, June). Actes de l'atelier APIMU 2021@ EIAH: Apprendre la Pensée Informatique de la Maternelle à l'Université. In *Atelier: Apprendre la Pensée Informatique de la Maternelle à l'Université (APIMU@ EIAH 2021)* (No. Atelier 11). HAL.
- Buitrago Flórez, F., Casallas, R., Hernández, M., Reyes, A., Restrepo, S., & Danies, G. (2017). Changing a generation's way of thinking: Teaching computational thinking through programming. *Review of Educational Research*, 87(4), 834-860.

- Carlborg, N., Tyrén, M., Heath, C., & Eriksson, E. (2019). The scope of autonomy when teaching computational thinking in primary school. *International journal of child-computer interaction*, 21, 130-139.
- Carretero Gomez, S., Vuorikari, R. and Punie, Y., DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use, EUR 28558 EN, Publications Office of the European Union, Luxembourg, 2017, ISBN 978-92-79-68006-9 (pdf),978-92-79-68005-2 (print),978-92-79-74173-9 (ePub), doi:10.2760/38842 (online),10.2760/836968 (print),10.2760/00963 (ePub), JRC106281.
- Casanova de Vilalta, L., Theophilou, E., Beardsley, M., Santos Rodríguez, P., & Hernández Leo, D. (2020). Distributed retrieval practice enhances primary school students' retention of computational thinking concepts. In *Sampson DG, Ifenthaler D, Isaías P, editors. 17th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2020); 2020 Nov 18-20; Lisbon, Portugal. Lisbon: IADIS Press; 2020 p. 189-96. IADIS Press.*
- Chen, G., Shen, J., Barth-Cohen, L., Jiang, S., Huang, X., & Eltoukhy, M. (2017). Assessing elementary students' computational thinking in everyday reasoning and robotics programming. *Computers & Education*, 109, 162-175.
- Chong, J., & Hurlbutt, T. (2007, May). The social dynamics of pair programming. In *29th International Conference on Software Engineering (ICSE'07)* (pp. 354-363). IEEE.
- Chou, P. N., & Shih, R. C. (2021, November). Young Kids' Basic Computational Thinking: An Analysis on Educational Robotics Without Computer.

- In *International Conference on Innovative Technologies and Learning* (pp. 170-180). Springer, Cham.
- D'Angelo, S., & Begel, A. (2017, May). Improving communication between pair programmers using shared gaze awareness. In *Proceedings of the 2017 CHI conference on human factors in computing systems* (pp. 6245-6290).
- da Cruz Alves, N., Gresse Von Wangenheim, C., & Hauck, J. C. (2019). Approaches to assess computational thinking competences based on code analysis in K-12 education: A systematic mapping study. *Informatics in Education*, 18(1), 17-39.
- del Olmo-Muñoz, J., Cózar-Gutiérrez, R., & González-Calero, J. A. (2020). Computational thinking through unplugged activities in early years of Primary Education. *Computers & Education*, 150, 103832.
- Demo, G.B., Marciano, G., Siega, S. (2008). Concrete Programming: Using Small Robots in Primary Schools. *Advanced Learning Technologies, ICALT '08*. Eighth IEEE International Conference on Advanced Learning Technologies, Santander - Cantabria , 1-5 july 2008, (p.301 – 302).
- Denis, B. (2020). Erasmus+ PIAF, un projet au service de la formation des encadrants d'activités visant développement de la Pensée Informatique et Algorithmique dès 5 ans [séminaire]. Université de Liège, Université de Lorraine, Universität des Saarlande, Université du Luxembourg. 81 https://piaf.loria.fr/wp-content/uploads/2020/02/Pr%C3%A9sentation-du-projet_29-janvier-2020_-_Brigitte-Denis.pdf
- Denis, B., & Vandeput, É. (2006). Le scénario pédagogique: outil d'expression des compétences TOP des enseignants. *Scénariser l'enseignement et l'apprentissage:*

une nouvelle compétence pour le praticien? Communication présentée à la 8e biennale de l'éducation, INRP, Lyon. In *8e Biennale de l'Education*.

Denner, J., Werner, L., Campe, S., & Ortiz, E. (2014). Pair programming: Under what conditions is it advantageous for middle school students?. *Journal of Research on Technology in Education*, 46(3), 277-296.

Denner, J., Campe, S., & Werner, L. (2019). Does computer game design and programming benefit children? A meta-synthesis of research. *ACM Transactions on Computing Education (TOCE)*, 19(3), 1-35.

Dongo, T. A., Reed, A. H., & T O'Hara, M. (2016). Exploring pair programming benefits for MIS majors. *Journal of Information Technology Education. Innovations in Practice*, 15, 223.

Doubé, W., & Lang, C. (2012). Gender and stereotypes in motivation to study computer programming for careers in multimedia. *Computer Science Education*, 22(1), 63-78.

Durak, H. Y., & Saritepeci, M. (2018). Analysis of the relation between computational thinking skills and various variables with the structural equation model. *Computers & Education*, 116, 191-202.

El Niño. (2018). *Computational thinking in primary education, why and how?* codeburst.io. <https://codeburst.io/computational-thinking-in-primary-education-why-and-howa2286cdff458>

European Education and Culture Executive Agency, Eurydice, *Chiffres clés de l'éducation et de l'accueil des jeunes enfants en Europe, 2019*, Publications Office, 2019, <https://data.europa.eu/doi/10.2797/067904>

- Faber, H. H., Wierdsma, M. D., Doornbos, R. P., van der Ven, J. S., & de Vette, K. (2017). Teaching computational thinking to primary school students via unplugged programming lessons. *Journal of the European Teacher Education Network, 12*, 13-24.
- González, M. R. (2015, July). Computational thinking test: Design guidelines and content validation. In *Proceedings of EDULEARN15 conference* (pp. 2436-2444).
- Greff, E. (1998), Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants, *Revue Sciences et techniques éducatives 5*(1), 47-61.
- Grover, S. et Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43.
<http://journals.sagepub.com/doi/abs/10.3102/0013189X12463051>
- Gurian, M., Stevens, K., & King, K. (2011). *Strategies for teaching boys and girls--secondary level: A workbook for educators*. John Wiley & Sons.
- Halpern, D. F., & Hakel, M. D. (2002). Learning that lasts a lifetime: teaching for long-term retention and transfer. *New Directions for Teaching and Learning, 2002*(89), 3-7.
- Hanks, B. (2008). Empirical evaluation of distributed pair programming. *International Journal of Human-Computer Studies, 66*(7), 530-544.
- Hennico, J., Reuter, R., & Weinberger, A. (2021). Factors Affecting the Implementation of Computational Thinking in the Curriculum. <http://hdl.handle.net/10993/48689>

- Henry, J., Lombart, C., & Dumas, B. (2021). Changer la représentation de l'informatique chez les jeunes: recommandations. In *Atelier «Apprendre la Pensée Informatique de la Maternelle à l'Université», dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH)* (pp. 13-24).
- Jobs, S. (1995). *Steve Jobs: The Lost Interview*.
- Kelleher, C., Pausch, R., & Kiesler, S. (2007, April). Storytelling alicie motivates middle school girls to learn computer programming. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1455-1464).
<https://doi.org/10.1145/1240624.1240844>
- King, K., & Gurian, M. (2006). The Brain--His and Hers. *Educational Leadership*, 64(1), 59.
- Klein, P. S., Nir-Gal, O., & Darom, E. (2000). The use of computers in kindergarten, with or without adult mediation; effects on children's cognitive performance and behavior. *Computers in Human Behavior*, 16(6), 591-608.
- Komis, V., & Misirli, A. (2013). Étude des processus de construction d'algorithmes et de programmes par les petits enfants à l'aide de jouets programmables. In *Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif*.
- Kommer, D. (2006). Boys and girls together: A case for creating gender-friendly middle school classrooms. *The Clearing House: A Journal of Educational Strategies, Issues and Ideas*, 79(6), 247-251.

- Korkmaz, Ö., Çakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in human behavior*, 72, 558-569.
- Lewis, C. M. (2011). Is pair programming more effective than other forms of collaboration for young students?. *Computer Science Education*, 21(2), 105-134.
- Liebenberg, J., Mentz, E., & Breed, B. (2012). Pair programming and secondary school girls' enjoyment of programming and the subject Information Technology (IT). *Computer Science Education*, 22(3), 219-236.
- Menon, D., Sowmya, B. P., Romero, M., & Viéville, T. (2020). Going beyond digital literacy to develop computational thinking in K-12 education. In *Epistemological Approaches to Digital Learning in Educational Contexts* (pp. 17-34). Routledge.
- Moore, T. J., Brophy, S. P., Tank, K. M., Lopez, R. D., Johnston, A. C., Hynes, M. M., & Gajdzik, E. (2020). Multiple representations in computational thinking tasks: a clinical study of second-grade students. *Journal of Science Education and Technology*, 29(1), 19-34.
- Moreno-León, J., & Robles, G. (2015, November). Dr. Scratch: A web tool to automatically evaluate Scratch projects. In *Proceedings of the workshop in primary and secondary computing education* (pp. 132-133).
- Nielsen, J. A., Zielinski, B. A., Ferguson, M. A., Lainhart, J. E., & Anderson, J. S. (2013). An evaluation of the left-brain vs. right-brain hypothesis with resting state functional connectivity magnetic resonance imaging. *PloS one*, 8(8), e71275.

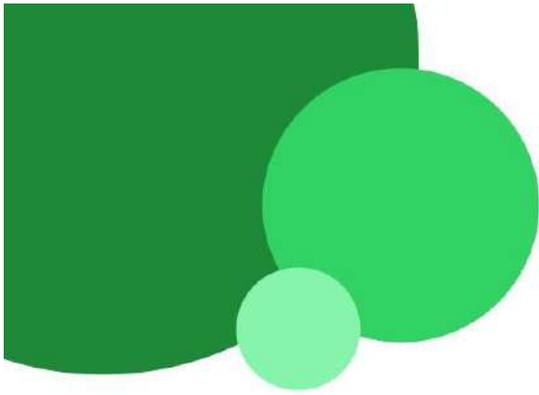
- Nogry, S. (2018). Comment apprennent les élèves au cours d'une séquence de robotique éducative en classe de CP?. *De 0 à 1 ou l'heure de l'informatique à l'école*, 235.
- Papavlasopoulou, S., Sharma, K., & Giannakos, M. N. (2020). Coding activities for children: Coupling eye-tracking with qualitative data to investigate gender differences. *Computers in Human Behavior*, 105, 105939.
- Papert, S. (1980). "Mindstorms" Children. *Computers and powerful ideas*.
- Parmentier, Y., Reuter, R., Higuete, S., Kataja, L., Kreis, Y., Dufлот-Kremer, M., ... & Denis, B. (2020, June). PIAF: developing computational and algorithmic thinking in fundamental education. In *EdMedia+ Innovate Learning* (pp. 315-322). Association for the Advancement of Computing in Education (AACE).
- Poulakis, E., & Politis, P. (2021). Computational thinking assessment: literature review. *Research on E-Learning and ICT in Education*, 111-128.
- Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, 29(4), 482-498.
- Resnick, M., & Siegel, D. (2015). A different approach to coding. *International Journal of People-Oriented Programming*, 4(1), 1-4.
- Resnick, M. (2007, June). All I really need to know (about creative thinking) I learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition* (pp. 1-6).
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior*, 72, 678-691.

- Román-González, M., Pérez-González, J. C., Moreno-León, J., & Robles, G. (2018). Can computational talent be detected? Predictive validity of the Computational Thinking Test. *International Journal of Child-Computer Interaction*, 18, 47-58.
- Romero, M. (2017). Les compétences pour le XXIe siècle. *Usages créatifs du numérique pour l'apprentissage au XXIe siècle*, 15-28.
- Romero, M., Lille, B., Viéville, T., Duflot-Kremer, M., de Smet, C., & Belhassein, D. (2018, August). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché. In *Educode-Conférence internationale sur l'enseignement au numérique et par le numérique*.
- Romero, M., Noirpoudre, S., & Viéville, T. (2018). Que disent les sciences de l'éducation à propos de l'apprentissage du code?. *Revue de l'EPI (Enseignement Public et Informatique)*.
- Selby, C. et Woollard, J. (2013). Computational thinking: the developing definition (Rapport). <https://eprints.soton.ac.uk/356481/>
- Shute, V. J., Sun, C. et Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://www.sciencedirect.com/science/article/pii/S1747938X17300350>
- Sigayret, K., Tricot, A., & Blanc, N. (2021). Pensée informatique et activités de programmation: quels outils pour enseigner et évaluer? In *Atelier «Apprendre la Pensée Informatique de la Maternelle à l'Université», dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH)* (pp. 68-75).

- Stevens-Smith, D., & Fleming, D. (2016). Learning preference changes in the educational setting. *International Journal of Assessment and Evaluation*, 23(2), 9-25.
- Stewart, W. H., Baek, Y., Kwid, G., & Taylor, K. (2021). Exploring Factors That Influence Computational Thinking Skills in Elementary Students' Collaborative Robotics. *Journal of Educational Computing Research*, 59(6), 1208-1239.
- Stoeckelmayr, K., Tesar, M., & Hofmann, A. (2011, September). Kindergarten children programming robots: a first attempt. In *Proceedings of 2nd International Conference on Robotics in Education (RIE)*.
- Strebelle, A., Melot, L., Ducarme, A., & Depover, C. (2017). Analyse des comportements sociaux dans le cadre d'un apprentissage collectif de la programmation d'un robot de sol. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation*, 24(1), 205-229.
- Sullivan, A., & Bers, M. U. (2016). Girls, boys, and bots: Gender differences in young children's performance on robotics and programming tasks. *Journal of Information Technology Education. Innovations in Practice*, 15, 145.
- Sun, L., Hu, L., & Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, 181, 104457.
- Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4), 295-312.
- Tchounikine, P. (2017). Initier les élèves à la pensée informatique et à la programmation avec Scratch. Consulté à l'adresse <http://ligmembres.imag.fr/tchounikine/PenseeInformatiqueEcole.html>.

- Tikva, C., & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, *162*, 104083.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.
- Wei, X., Lin, L., Meng, N., Tan, W., & Kong, S. C. (2021). The effectiveness of partial pair programming on elementary school students' computational thinking skills and self-efficacy. *Computers & Education*, *160*, 104023.
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012, February). The fairy performance assessment: Measuring computational thinking in middle school. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 215-220).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Yadav, A., Hong, H., & Stephenson, C. (2016). Computational thinking for all: Pedagogical approaches to embedding a 21st century problem solving in K-12 classrooms. *TechTrends*, *60*, 565–568. doi:[10.1007/s11528-016-0087-7](https://doi.org/10.1007/s11528-016-0087-7).
- Yang, Y. F., Lee, C. I., & Chang, C. K. (2016). Learning motivation and retention effects of pair programming in data structures courses. *Education for Information*, *32*(3), 249-267.
- Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2020, April). Computational thinking test for beginners: Design and content validation. In *2020 IEEE Global Engineering Education Conference (EDUCON)* (pp. 1905-1914). IEEE.

- Zapata-Cáceres, M., Martín-Barroso, E., & Román-González, M. (2021). Collaborative Game-Based Environment and Assessment Tool for Learning Computational Thinking in Primary School: A Case Study. *IEEE Transactions on Learning Technologies*, 14(5), 576-589.
- Zhang, Y., Luo, R., Zhu, Y., & Yin, Y. (2021). Educational robots improve K-12 students' computational thinking and STEM attitudes: systematic review. *Journal of Educational Computing Research*, 59(7), 1450-1481.
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2017). Investigating the period of switching roles in pair programming in a primary school. *Journal of Educational Technology & Society*, 20(3), 220-233.



Au cours des dix dernières années, l'apprentissage du numérique a concouru au changement de l'enseignement et ce, par l'utilisation d'une multitude d'approches pédagogiques améliorées par la technologie qui a permis notamment d'aider à la compréhension des concepts de base de la pensée informatique (Menon, 2019). Malgré le fait que les chercheurs aient mis l'accent sur l'importance de développer la pensée informatique et algorithmique chez les enfants et ce dès l'école maternelle, l'étude de son développement chez les jeunes enfants n'en est qu'à ses balbutiements (Bers et al, 2014). Si nous nous référons au rapport Eurydice, publié en 2019, évaluant les politiques éducatives en matière de numérique, nous pouvons en ressortir le fait que la Belgique francophone fait partie des « mauvais élèves » en termes de développement de ses compétences numériques tant au niveau primaire que secondaire. La réforme du Pacte pour un Enseignement d'excellence devrait néanmoins pouvoir assouvir ces manquements (Henry et al., 2021). Le but de cette recherche poursuit deux objectifs. Le premier réside dans le fait de concevoir un scénario pédagogique qui permettrait de développer les compétences relatives à la pensée informatique et algorithmique (PIA) chez des élèves du premier cycle de l'enseignement primaire de la Fédération Wallonie Bruxelles. Le second, dans la lignée des recherches menées par Angeli et Valanides en 2020, est d'y intégrer deux procédures d'étayage différentes, de voir si l'implémentation de l'une ou l'autre technique engendre une différence et si celle-ci se veut genrée.

