

Former les enseignants à et par la programmation informatique cocreative en formation initiale

10

Frédérique LONGUET¹
Marcela Georgina GÓMEZ-ZERMEÑO²

1. Sorbonne Université, INSPE de l'Académie de Paris, 75017 Paris, France

2. Institut de recherche, d'innovation et d'études supérieures pour l'éducation, État de Nuevo León, Mexique

Ces dernières années, la majorité des gouvernements du monde entier se sont penchés sur l'intégration de la technologie comme un moyen de changement éducatif. Ceux qui promeuvent une plus grande utilisation du numérique dans l'éducation ont imaginé des écoles où les élèves acquièrent les « compétences du XXI^e siècle » de manière personnalisée et collaborative, et à leur propre rythme avec des enseignants jouant le rôle de facilitateurs d'apprentissage (Voogt et Roblin, 2012). Or, l'éducation n'est pas seulement une question technique, mais aussi une clé pour vivre dans un monde en perpétuelle mutation. Pour comprendre les enjeux et soutenir une vision transformatrice de l'éducation, il nous semble nécessaire d'adopter une approche historique et contextualisée, en évitant le déterminisme technologique et le plaidoyer mal informé pour un passage d'une pédagogie anachronique et « analogique » à une pédagogie innovante et « numérique », fondée sur la reliance et le milieu (Longuet et Springer, 2021). C'est pourquoi nous sommes intéressées à une approche systémique de la programmation cocreative en formation initiale des enseignants. La programmation cocreative également appelée codéveloppement est une approche dans laquelle les programmeurs travaillent ensemble sur la création d'un programme informatique (*mob programming*). Cette approche diffère de la programmation individuelle traditionnelle, où un seul développeur écrit tout le code. Les enseignants stagiaires programmeurs collaborent étroitement, partagent des idées, résolvent des défis autour d'un seul écran. Cette programmation en équipe visant l'exploitation de la synergie entre les enseignants stagiaires conduit à des propositions didactiques et pédagogiques diverses en fonction des situations et des histoires personnelles.

Considérée comme une littérature du XXI^e siècle, la programmation informatique est intégrée depuis 2016 dans les programmes scolaires en France. Bien que la programmation informatique occupe

une place significative dans le champ scolaire du cycle 1 à la terminale, elle ne s'inscrit pas dans une didactique de l'informatique dont les savoirs et savoir-faire seraient à maîtriser en formation initiale des enseignants. Elle est d'une part essentiellement arriérée aux mathématiques, aux sciences et à la technologie (Romero, 2016) et d'autre part définie dans le socle commun de connaissances, de compétences et de culture comme un langage parmi d'autres pour penser, partager des visions du monde (domaine 1 du socle) lors de situations cocreatives (domaine 2 et 3 du socle).

Des compétences transversales comme la créativité, la collaboration, la résolution de problèmes et la pensée critique sont de ce fait associables à la programmation pour que celle-ci ne soit pas réduite à des habiletés techniques (Romero *et al.*, 2017). La programmation cocreative semble donc être une clé pour répondre aux enjeux de l'éducation en termes de complexité, de diversité, d'interdisciplinarité et de créativité et une stratégie pour engager les enseignants et leurs élèves novices du 1^{er} et 2^e degrés issus de différentes disciplines dans une pensée complexe qui dépasse la programmation procédurale.

Dans ce cadre, notre question de recherche peut être formulée ainsi : quel est le rôle de la cocreativité dans le développement de la pensée complexe, dans l'acquisition de connaissances et compétences informatiques et transversales en formation initiale des enseignants ?

Former à et par la programmation cocreative

Opter pour une approche expérientielle

Considérer la programmation cocreative comme une ouverture à l'expérience en formation professionnelle, c'est s'inscrire à la fois dans une vision philosophique existentialiste qui considère la construction de l'identité personnelle et professionnelle comme une démarche active et anthropologique qui place l'être humain au cœur de la formation. Pour Dorais (2004), l'identité n'est pas pré-donnée, fixée ou extérieure mais construite à travers la confrontation personnelle du sujet à son environnement et aux autres. Elle est à la fois personnelle puisqu'elle résulte de l'intention du sujet à entrer en relation avec son environnement, et collective puisque les interactions avec l'environnement supposent le dialogue avec les autres sujets. La construction de l'identité personnelle et professionnelle est donc liée à la capacité de traiter avec le réel, capacité à l'origine de toute connaissance selon Aristote (1840), de dialoguer, de négocier. L'expérience en tant que contact avec le monde complexe dans lequel le sujet évolue, est donc fondamentale. Pour Vygotski, les expériences émotionnellement vécues appelées *perezhivanie* (перезивание, « au-delà » en russe), sont des zones

de développement des potentialités (Veresov, 2014). Elles sont considérées comme un processus de transformation indispensable au développement de la personne. L'identité personnelle et professionnelle se développe donc au sein d'une communauté grâce aux expériences médiées par les artefacts, le dialogue collaboratif et la sémiotique sociale (Longuet et Springer, 2021). Tout développement humain est « conjoint des autonomies individuelles, des participations communautaires et de la conscience d'appartenir à l'espèce humaine » (Morin, 2014, p. 105). L'expérience créative dialoguée ouvre la voie vers la connaissance de soi, de ses potentialités, du monde. En s'appropriant les artefacts, les enseignants s'énaquent (Maturana et Varela, 1994) et créent une culture partagée renouvelée.

Penser la complexité

La programmation cocréative permet d'engager les enseignants dans un processus dialogique de recherche création pédagogique (Longuet, 2014). Elle contraint les enseignants à réaliser une production observable nouvelle adaptée à la situation (Bonnardel et Lubart, 2019). Elle crée ainsi les conditions de la recherche, de l'exploration, du dialogue qui cherche la compréhension des choses. Elle oblige les enseignants à se questionner, à identifier les situations problématiques inhérentes à la contrainte de programmation, à accepter le désordre, à affronter les incertitudes, à peser les contradictions et les complémentarités pour agir et arriver à une création acceptable. La création contraint les enseignants à « ne pas se contenter des modes d'emploi existants », à « se rejoindre dans une certaine abstraction prospective » (Rouquette, 1973, p. 122). Elle les oblige à développer une intelligence stratégique qui leur offre un certain nombre de scénarios possibles pour l'action qui peuvent être modifiés au cours du processus de création (Morin, 2014). La pensée stratégique relie de ce fait les enseignants ; elle les contraint à « langager », c'est-à-dire à se coordonner avec d'autres, à s'ajuster à d'autres pensées, à opérer dans le langage pour externaliser les problèmes, prendre des décisions, faire progresser la création, construire des connaissances (Longuet et Springer, 2021). La médiation est au cœur du processus dynamique de création. Le pôle du « langager » est central. C'est en quelque sorte le moteur de la pensée complexe. Il est essentiel pour agir. Il met en mouvement la pensée.

Développer des compétences dans une approche par situations

L'Organisation des Nations unies pour l'éducation, la science et la culture (UNESCO) reconnaît que la prévalence des technologies de l'information et de la communication (TIC) est potentiellement

importante dans l'éducation et a développé le cadre de compétences TIC de l'UNESCO pour les enseignants (UNESCO, 2018), car il s'agit d'un outil utile pour guider la formation initiale et la recherche d'enseignants dans l'éducation. Avec ce référentiel, l'UNESCO vise à fournir une base pour l'élaboration de politiques actualisées et le renforcement des capacités dans le domaine dynamique des TIC. Le *Référentiel UNESCO de compétences TIC pour les enseignants* (2018) met en évidence le rôle que la technologie peut jouer dans le soutien de six domaines clés de l'éducation à travers trois phases d'apprentissage. Il est une référence pour les développements technologiques et pédagogiques récents dans les domaines des TIC et de l'éducation. Il souligne l'importance pour les enseignants de favoriser chez les élèves les compétences de collaboration, de résolution de problèmes et de créativité dans l'utilisation des technologies numériques (UNESCO, 2018).

À l'ère numérique, ces compétences deviennent partie intégrante de leur formation initiale (Montiel et Gómez-Zermeño, 2022). Elles peuvent être développées dans des situations de programmation cocreative. Pour cela, Wilson *et al.* (2012) proposent un cadre pour établir le niveau de compétence en programmation cocreative, en analysant un projet collaboratif en termes de concepts de programmation (tels que les fils d'exécution parallèles (*threads*), les instructions conditionnelles ou les variables), d'organisation du code (noms de variables, noms d'objets et blocs étranges), et de conception et de convivialité (telles que la fonctionnalité ou l'originalité, entre autres).

Les compétences documentées par l'UNESCO et les chercheurs sont dans notre cas construites, actualisées par les enseignants en situation de programmation cocreative. Elles émergent du processus de création. Jonnaert (2017, p. 44) considère qu'« une compétence se construit par des personnes en situations. Une compétence caractérise un moment particulier, celui de l'harmonie entre ces personnes et ces situations, le moment qui leur permet d'affirmer que leurs actions dans ces situations sont viables à cet instant. » Cette définition rappelle le caractère situé de la compétence, non prédéfini et l'importance de la relation entre les personnes. Les enseignants développent, complètent, transforment leur répertoire professionnel dans le cours de l'action dialoguée médiée par des artefacts matériels (logiciel de programmation éducative, robots éducatifs, etc.) et symboliques (langage de programmation, etc.). Leur curriculum se construit au fur et à mesure de la cocreation. La démarche choisie s'écarte de la démarche classique de la preuve étayée par le référentiel institutionnel. Les enseignants enrichissent leur répertoire professionnel didactique et pédagogique dans le cours de l'action et traduisent leur capital compétences dans un projet développé pour leurs élèves.

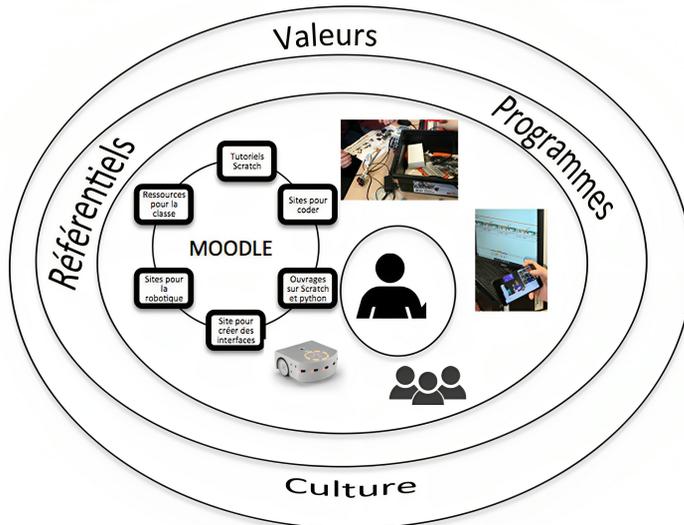


FIGURE 1 – Écosystème d’apprentissage et de recherche développé par le formateur INSPE.

Dispositif de formation

Un écosystème d’apprentissage et de recherche hybride

Le dispositif de formation est conçu à partir du modèle de Bronfenbrenner (1979). Il est écologique et sociosémiotique. Il favorise la reliance, c’est-à-dire la nécessité de se relier à soi et aux autres pour saisir des différences, des valeurs particulières, une compréhension du monde. Les enseignants évoluent dans un système environnemental complexe, systémique allant d’un microsystème à un macrosystème dans lequel chaque système communique avec un système plus vaste, renforçant ainsi la relation entre l’individu et son environnement comme le montre le schéma de la figure 1.

Les étudiants de l’INSPE disposent de la plateforme Moodle sur laquelle des ressources ont été déposées pour éclairer la contrainte de programmation cocréative. Le formateur a créé des liens référencés dans le tableau 1, vers des tutoriels pour apprendre à utiliser Scratch, des sites pour apprendre à coder comme Swift Playgrounds ou Algoblocs, des sites pour la robotique comme Thymio cycle 1.2.3, Lego ou Drones et robots à l’école, des ouvrages sur Scratch et Python et des ressources pour la classe comme Blockly games, Scratch exploreur studio.code.org/courses et Python lycée.

La plateforme Moodle est utilisée par les étudiants en présentiel et à distance. Le formateur a également mis à disposition

TABLEAU 1 – Liens référencés par le formateur.

Environnement et développement	https://scratch.mit.edu/ideas http://www.scratchjr.org/ https://www.apple.com/swift/playgrounds/ https://www.algoblocs.fr/ https://repl.it/languages/python3
Ressources pour la classe	http://www.reseau-canope.fr/atelier-yvelines/spip.php?article1161 https://blockly.games/ https://www.pedagogie.ac-aix-marseille.fr/jcms/c_-10544187/fr/scratch-explorateur http://www.kidscod.in/ https://studio.code.org/courses http://python.lycee.free.fr/
Ouvrages sur Scratch, Python	https://www.eyrolles.com/Recherche/?q=scratch&Rayon[INF] https://www.eyrolles.com/Recherche/?q=python+kids
Robotique	https://www.terrapiologo.com/products/robots/bee-bot/bee-bot.html cycle 1 https://www.thymio.org/fr/ cycle 2 1 3 https://www.lego.com/en-us/themes/mindstorms cycle 4 https://paper.li/RfrentNumrique/1454506123#/ drones et robots
Création d'interfaces	https://makeymakey.com/

des enseignants des kits Lego de construction de robots ainsi que des robots BeeBot. Cet environnement est augmenté par les étudiants s'ils le souhaitent. Certains mobilisent en présentiel pendant l'expérience cocreative des smartphones et des ordinateurs portables.

L'expérience de programmation cocreative s'inscrit dans une dynamique perception-action dans la mesure où les enseignants explorent l'environnement, sélectionnent ce qui est essentiel pour leur développement, voire l'enrichissent. Ils créent des partitions multiples, des *umwelts* (« environnements » en allemand; Von Uexküll, 2010) à chaque moment expérimental. Le formateur n'intervient pas dans le processus de cocreation. Il présente seulement l'écosystème d'apprentissage et de recherche lors de la première séance et il participe à la coévaluation finale en fin de TD.

Présentation du public

La formation à la programmation cocreative s'est déroulée à l'INSPE Paris Sorbonne entre 2017 et 2019 dans le cadre de l'UE « projets pour la classe » du tronc commun structuré autour de quatre ensembles : les valeurs de la République, des gestes professionnels liés aux situations d'apprentissage et d'éducation, une

appropriation des thèmes transversaux d'éducation. Quarante-cinq étudiants du master MEEF, de l'éducation et de la formation, 1^{er} et 2^e degrés toutes disciplines confondues, ont suivi une formation d'une durée de 21 heures. L'objectif était d'initier des étudiants novices à l'informatique, à et par la programmation cocréative pour qu'ils construisent la compétence de programmation mais aussi pour qu'ils développent d'autres compétences en rapport avec le référentiel des enseignants comme la collaboration, la construction, la mise en œuvre et l'animation de situations d'enseignement et d'apprentissage prenant en compte la diversité des élèves.

Démarche expérientielle : défi et contrainte de création

Le formateur a choisi de confronter les enseignants à trois défis puis à une contrainte de création pédagogique pour les engager dans un processus de création sans modèle de résolution les obligeant à explorer l'écosystème, à penser stratégiquement, à entrer dans une conversation réflexive schöniennienne (Longuet, 2018) pour orienter leurs choix comme le montre la figure 2.

Les enseignants relèvent trois défis. Le premier consiste à créer un jeu permettant à des élèves de travailler les tables de multiplication :

Vous devez créer un jeu permettant à vos élèves de travailler les tables de multiplication. Un personnage pose les questions à vos élèves. L'élève commence avec cinq vies. Lorsque toutes les vies sont écoulées, un deuxième personnage vient manger le personnage principal. Les élèves doivent avoir des retours sur la validité ou non de leur réponse et le nombre de bonnes réponses.

Le second défi consiste à faire réaliser aux élèves un jeu permettant à un animal de se déplacer dans un labyrinthe pour atteindre une forme rouge en 45 secondes :

Le déplacement de l'animal se fait avec les flèches du clavier de l'ordinateur. À chaque fois que l'animal touche un bord du labyrinthe, il perd une vie. Au bout de cinq vies perdues, il revient au début. Une animation finale indique si le joueur a perdu ou gagné. Nombre de lignes de blocs : entre 35 et 40.

Le troisième défi consiste à réaliser un jeu sur les nombres pairs et impairs.



FIGURE 2 – Exploration de l'écosystème par les étudiants.

Vos élèves doivent réaliser un jeu sur les nombres pairs et impairs. Des nombres un à neuf descendent automatiquement et les élèves doivent cliquer sur ceux qui sont pairs (ou impairs). Progressivement les nombres descendent de plus en plus rapidement. Une vie est supprimée en cas d'erreur si le nombre n'a pas été cliqué en arrivant en bas. Les élèves ont un retour sur le nombre de vies et le nombre de bonnes réponses. Indice : il faut utiliser la fonction de clonage. Nombre de lignes de blocs : entre 27 et 30.

Ces défis les amènent à faire face à diverses situations de programmation cocréative en analysant un projet collaboratif en termes de concepts de programmation, de développement de leurs compétences de collaboration, de résolution de problèmes et de créativité dans le cours de l'action dialoguée médiée par des artefacts matériels et symboliques. Ils disposent ainsi d'un réservoir de solutions qu'ils pourront réutiliser ou explorer à nouveau, faire progresser.

Une fois les défis relevés, les enseignants conçoivent deux projets. Le premier consiste à concevoir un projet dans lequel les élèves développent un jeu avec Scratch. Le second consiste à concevoir un projet avec le robot Thymio ou Lego. Dans les deux cas, les enseignants stagiaires doivent remettre une fiche descriptive du projet, un exemple simulé du projet réalisé par un groupe d'élèves, la fiche de programmation sauvegardée depuis le logiciel, la photo d'un exemple simulé d'interface développée par les élèves et la vidéo de démonstration de l'activité réalisée avec le robot.

Méthodologie de la recherche

Une recherche située compréhensive

Cette étude s'inscrit dans une démarche de recherche située, compréhensive (Dayer et Charmillot, 2012). Elle vise à décrire et interpréter les dynamiques des participants. La réflexion inductive ancrée dans le paradigme d'exploration (Guillemette et Luckerhoff, 2012) permet une construction progressive du sens faisant apparaître des éléments qui en fin de compte font système. Ce type de recherche accorde une place significative au vécu des acteurs. Cela implique que l'on accepte l'idée que ces acteurs disposent d'une rationalité et d'une intentionnalité réelles et qu'ils ont la volonté de la partager et de dialoguer avec des partenaires (Longuet, 2018). Les traces, esquisses, photos, vidéos qui se déploient dans Moodle constituent le corpus de notre recherche. Elles dévoilent l'action stratégique, les prises de décision, le développement progressif d'acquis conceptuels.

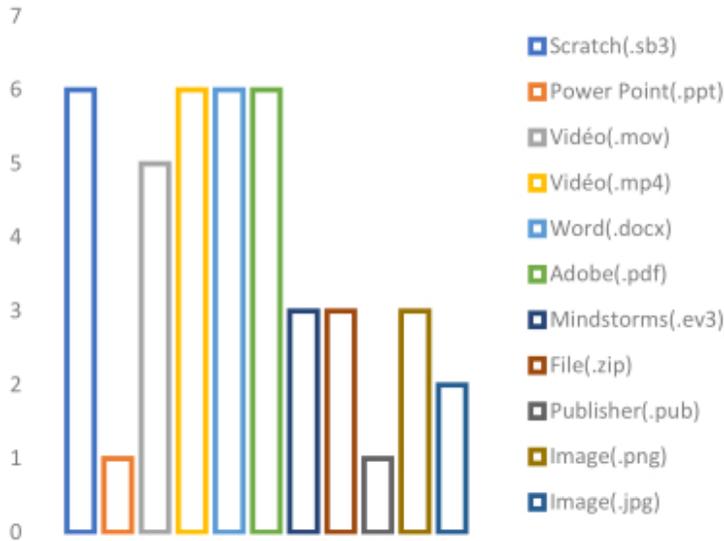


FIGURE 3 – Fichiers informatiques du portfolio électronique.

Nous considérons que cette démarche est pertinente, car il y a un déficit d’outils qui aident les enseignants à évaluer le développement de la pensée computationnelle (PC) et la correction des projets collectifs programmés (Montiel et Gómez-Zermeño, 2021; Wing, 2008). Concernant le langage de programmation Scratch, plusieurs auteurs ont proposé différentes approches pour évaluer le développement de la PC en analysant les projets des apprentis, mais la majorité de ces approches se basent exclusivement sur une analyse manuelle (Moreno *et al.*, 2015).

Collecte des données

Brennan et Resnick (2012) ont introduit une stratégie basée sur l’analyse de portefeuilles de projets à l’aide d’un outil de visualisation appelé Scrape (Wolz *et al.*, 2011), complétée par des interviews sur les projets et la scénographie. Dans le but d’appliquer une démarche de recherche située dans la présente étude, les données générées par chacun des groupes de travail collaboratif ont été recueillies dans un portfolio électronique. La figure 3 montre à quels types de fichiers informatiques les données collectées correspondent.

Analyse des traces, manifestations de l’activité cocréative des enseignants

Suivant les recommandations de Moreno *et al.* (2015), la démarche de recherche située a été appliquée à travers l’analyse des dossiers inscrits au portfolio de chacun des groupes d’enseignants qui ont réalisé un travail collectif sur la programmation informatique

cocréative en formation initiale. Pour cela, deux techniques d'analyse ont été appliquées, la première basée sur une méthode Delphi et la seconde appuyée par des outils d'analyse des codes Scratch programmés par chaque groupe d'enseignants stagiaires.

Méthode Delphi

Conçue en 1950 par Olaf Helmer à la Rand Corporation (Okoli et Pawlowski, 2004), la méthode Delphi a pour but de mettre en évidence des convergences d'opinions et de dégager certains consensus sur des sujets précis, souvent avec un caractère prospectif important, grâce à la consultation d'experts. Aujourd'hui, la méthode Delphi est de plus en plus utilisée par plusieurs chercheurs en développement des modèles dans les domaines des systèmes d'information, de la gestion des connaissances et des technologies de l'information (TI), spécifiquement en développement des capacités en gestion de projets en TI.

Pour appliquer la méthode Delphi, une équipe d'experts a été constituée et composée de deux docteurs en innovation pédagogique, d'un docteur en éducation et TIC (*e-learning*) et deux ingénieurs en systèmes informatiques.

Dans le but d'analyser les fichiers que les enseignants ont enregistrés dans leurs portfolios, l'équipe d'experts a sélectionné un ensemble de compétences et sous-compétences du cadre de compétences en TIC proposé par l'UNESCO ; ils ont complété cette liste en s'appuyant sur les idées théoriques sur la pensée complexe, appliquées à la programmation cocréative.

Lors de l'analyse, les experts ont utilisé un guide d'observation composé de la liste des compétences et sous-compétences sélectionnées, pour évaluer les trois niveaux établis dans le but de générer des informations qui leur permettraient d'interpréter la dynamique générée par les groupes d'enseignants au cours de leur travail collaboratif de programmation cocréative.

Ces trois niveaux de compétences sont basés sur le cadre de compétences TIC pour les enseignants de l'UNESCO (UNESCO-ICTCFT) :

- Niveau 1 : acquisition de connaissances ;
- Niveau 2 : approfondissement des connaissances ;
- Niveau 3 : création de connaissances.

Le guide d'observation (tableau 2), a été développé avec les experts sur la base des idées théoriques des auteurs référencés dans cette étude. Les trois colonnes de droite de ce tableau correspondent aux trois niveaux explicités ci-dessus.

TABLEAU 2 – Guide d’observation des experts.

Compétences	Sous-compétences	Niveau 1	Niveau 2	Niveau 3
Programmation informatique	Traitement de l’information			
	Résolution de problèmes			
	Créativité			
	Concevoir des systèmes			
	Complexité			
	Pensée procédurale			
Programmation cocréative	Pensée informatique			
	Contrainte de création			
	Collaboration			
	Médiations			
	Expérience partagée			
	Langager			
Pensée complexe	Sémiotique sociale			
	Processus de cocréativité			
	Affronter les incertitudes			
	Peser les contradictions et complémentarités			
	Outils complexes			
	Écosystème d’apprentissage			
	Démarche expérientielle			
Pensée critique				
Pensée stratégique				

Dr. Scratch

Boe *et al.* (2013) ont développé Hairball, un analyseur de code statique qui détecte les problèmes potentiels dans les programmes, tels que le code qui ne s’exécute jamais, les messages qu’aucun objet ne reçoit ou les attributs qui ne sont pas correctement initialisés. Basé sur Hairball, Dr. Scratch est une application Web qui permet d’analyser automatiquement les projets Scratch pour vérifier s’ils ont été programmés correctement et recevoir des commentaires pour améliorer leur code, et développer leur capacité de PC.

Dr. Scratch détecte certaines mauvaises habitudes de programmation ou bogues potentiels, tels que l’utilisation de noms sans signification, le code redondant, le code qui ne s’exécute pas et l’initialisation incorrecte des attributs d’objet. Pour attribuer le score PC, Dr. Scratch déduit la capacité de réflexion computationnelle démontrée par le programmeur en fonction des sous-compétences (Moreno *et al.*, 2015).

1. Parallélisme : développer un programme qui exécute plusieurs actions en même temps.
2. Pensée logique : reconnaître le problème à résoudre, trouver une solution à programmer et analyser les résultats obtenus pour en tirer des conclusions.
3. Algorithmes de contrôle de flux : contrôlent les instructions ou les blocs qui effectuent les différentes actions.
4. Interactivité avec l'utilisateur : développer des programmes qui permettent aux utilisateurs de se sentir intégrés au jeu vidéo, à l'histoire ou au projet que nous créons, en travaillant d'une manière ou d'une autre.
5. Abstraction et décomposition des problèmes : développer des compétences de programmation complexes qui permettent de diviser un gros problème (ou un défi) en petits problèmes faciles à résoudre.
6. Représentation des informations : identifier l'ensemble des données et variables qui ont une valeur à chaque instant de l'exécution du programme, et qui peuvent être modifiées par les instructions ou la séquence de blocs.
7. Synchronisation : permet à un programme de suivre un certain ordre et d'effectuer une action lorsqu'un autre a terminé, formant une chaîne ordonnée d'actions.

Tous les programmes réalisés par les étudiants dans leur travail de cocréation collaborative ont été analysés avec Dr. Scratch. Il a été observé que la sous-compétence de l'interactivité avec les utilisateurs a obtenu les résultats les plus faibles, tandis que le contrôle de flux et l'abstraction ont obtenu les résultats les plus élevés. Ces résultats témoignent des compétences qu'ils ont développées pour résoudre des problèmes complexes et l'utilisation d'outils complexes.

De leur côté, les experts ont passé en revue les portfolios et les codes de chacun des programmes Scratch développés par les groupes d'étudiants. Ainsi, ils ont rempli un guide d'observation (tableau 2) pour chaque groupe, indiquant le niveau de sous-compétence qu'ils ont considéré, en fonction de leur expérience en programmation informatique cocréative et de leur connaissance de Scratch, lors de l'évaluation de chaque portfolio.

Les résultats montrent que les enseignants stagiaires novices en programmation ont développé des compétences en programmation informatique grâce à la programmation Scratch. Ils ont néanmoins atteint un stade de développement plus important en créativité et en pensée complexe. La programmation informatique a participé au développement de la pensée complexe et de la créativité.

L'analyse des portfolios a également révélé des imaginaires pluriels bien que marqués par les référentiels et les programmes en fonction

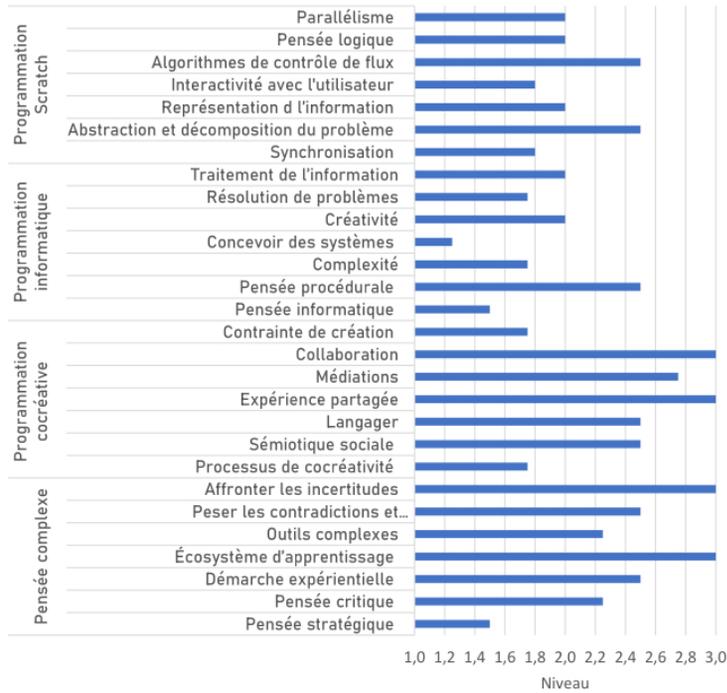


FIGURE 4 – Résultats de l'analyse avec la méthode Delphi et avec l'outil Dr. Scratch.

des milieux traversés par les enseignants stagiaires, de leur héritage culturel et de leurs valeurs. Elle a dévoilé le processus de stylisation professionnelle (Clot et Faïta, 2000) nécessaire au développement professionnel.

L'analyse confirme l'influence de l'écosystème sur le développement des personnes. Certains groupes ont perçu la programmation cocreative comme un moyen de transfert des apprentissages disciplinaires et de vérification des acquis. Le projet a servi par exemple à vérifier des acquis en histoire en cycle 3 (développement d'une conscience historique) ou en français au collège (structure du conte). D'autres groupes ont pensé la programmation cocreative comme un moyen d'apprentissage, un moyen par exemple de passer de l'espace vécu à l'espace représenté. D'autres groupes ont imaginé la programmation cocreative comme un levier pour vivre ensemble, inclure. Certains projets ont été développés pour favoriser la liaison CM2/6^e. Enfin, plusieurs groupes ont considéré la programmation comme un jeu. Les enseignants stagiaires ont exploré un environnement et sélectionné ce qui leur a semblé nécessaire pour leur développement professionnel. Chaque groupe a joué sa partition comme l'explique Uexküll (2010) à travers sa métaphore musicale. Chaque groupe constitué de compositeurs/programmeurs a orchestré un morceau particulier, tissé des liens pour se développer personnellement et professionnellement, s'inscrire dans un projet sociétal d'éducation.

Discussion et conclusion

Notre recherche s'est intéressée à une formation innovante à et par la programmation cocréative en rupture avec les formations transmissives (Romero *et al.*, 2017) et individuelles. La formation observée fondée sur l'expérience partagée et la conception d'un écosystème d'apprentissage et de recherche ouvert a embrassé la complexité pour favoriser l'émergence des zones de potentialité de chaque enseignant stagiaire.

Les résultats montrent que la stylisation professionnelle liée à l'action créatrice partagée a un impact important sur le développement de la compétence de programmation et les compétences du référentiel enseignant. Il semble de ce fait intéressant de considérer les enseignants stagiaires comme des acteurs créatifs curieux disposant d'un répertoire pluriel, de favoriser les opérations complexes pour qu'ils s'approprient les genres professionnels et les transforment, qu'ils puissent se projeter dans le monde professionnel dans lequel ils évoluent plutôt que d'être réduits à imiter les genres. Le passage à une pédagogie fondée sur la reliance, l'écologie, l'innovation nécessite néanmoins une transformation profonde des postures professionnelles des formateurs et des enseignants. La salle de formation et de classe doit se transformer en un véritable laboratoire social dans lequel les différentes partitions peuvent s'exprimer. Par ailleurs, les résultats de cette recherche mettent en évidence la nécessité d'intégrer des stratégies de développement des compétences en programmation informatique cocréative dans la formation initiale étant donné qu'elles permettent le développement de compétences qui renforcent la pensée complexe. Pour les recherches futures, il est recommandé de poursuivre l'analyse des stratégies pour impliquer les enseignants débutants dans l'utilisation d'outils qui vont au-delà de la programmation procédurale, dans le but de développer une pensée complexe. Une approche par situations de programmation cocréative de plus en plus complexes en formation initiale des enseignants pourrait être une piste, la compétence se situant dans un continuum.

En effet, la programmation cocréative est un outil fondamental pour faire face aux défis de l'éducation en termes de complexité, de diversité, d'interdisciplinarité et de créativité, qui permet de résoudre des problèmes complexes par la pensée systémique. Comprendre la complexité de la réalité vécue par les enseignants débutants dans les écoles, souscrit à la fois à une vision philosophique existentialiste qui considère la construction de l'identité personnelle et professionnelle comme un atout, et à une anthropologie qui place l'être humain au centre de sa formation.

Cette étude a été réalisée dans le cadre de la collaboration offerte par le projet pluridisciplinaire « Informatique à l'école : conceptualisations, accompagnement, ressources » (IE CARE), qui a

permis l'association de chercheurs internationaux reconnus pour leurs contributions dans les domaines des sciences humaines, des sciences sociales et des TIC. Leurs résultats corroborent d'une part avec la nécessité d'identifier les convergences des travaux de recherche dans le domaine des sciences humaines, sociales et des TIC; comprendre, évaluer et valider des scénarios pédagogiques et des ressources pédagogiques numériques; et d'autre part de construire un cadre de stratégies et d'indicateurs dans le but de développer des cultures numériques pour la formation des enseignants à l'innovation pédagogique.

Recommandations

Pour instaurer une formation à et par la programmation cocréative, il paraît nécessaire de passer d'une pédagogie centrée sur l'objet programmation à une pédagogie de la relation, écologique et sociosémiotique basée sur le partage des connaissances, la confrontation des approches, la multiplication des regards. Pour ce faire, il convient d'engager les apprenants au sein d'écosystèmes d'apprentissage et de recherche favorisant la reliance et le dialogue collaboratif (langager), notamment à travers l'imposition de contraintes créatives.

Références

- Alemán de la Garza, L., Bruillard, E., Gómez-Zermeño, M., *et al.* (2019). *Rethinking pedagogy : Exploring the potential of digital technology in achieving quality education*, UNESCO-UNESDOC Mahatma Gandhi Institute of Education for Peace and Sustainable Development (MGIEP). <https://unesdoc.unesco.org/ark:/48223/pf0000372786>
- Aristote (1840). *Métaphysique, Livre I.* (Victor Cousin trad., texte originel publié au IV^e siècle av. J.-C.). <http://remacle.org/bloodwolf/philosophes/Aristote/metaphysique1fr.htm>
- Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., et Franklin, D. (2013). Hairball : Lint-inspired static analysis of scratch projects, *Proceedings of the 44th ACM Technical Symposium on Computer Science Education (SIGCSE 2013)*, p. 215-220. <https://doi.org/10.1145/2445196.2445265>
- Bonnardel, N., et Lubart, T. (2019). La créativité : approches et méthodes en psychologie et en ergonomie, *RIMHE : Revue interdisciplinaire management, homme et entreprise*, vol. 37, n° 8, p. 79-98. <https://doi.org/10.3917/rimhe.037.0079>
- Brennan, K., et Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking, *Proceedings of the 2012 Annual Meeting of the American Educational Research*

Association (AERA 2012), vol. 1. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>

Bronfenbrenner, U. (1979). *The Ecology of Human Development : Experiments by Nature and Design*, Harvard University Press. <https://doi.org/10.4159/9780674028845>

Bruillard, É., et Fluckiger, C. (2018). Une approche didactique de l'informatique scolaire, *Revue française de pédagogie, recherche en éducation*, vol. 204, p. 86-87. <https://doi.org/10.4000/rfp.8523>

Clot, Y., et Faïta, D. (2000). Genres et styles en analyse de travail. Concepts et méthodes, *Travailler*, vol. 4, p. 7-42.

Dayer, C., et M. Charmillot (2012). Démarche compréhensive et méthodes qualitatives : clarifications épistémologiques, *Formations et pratiques d'enseignement en questions*, vol. 14, p. 163-176

Dorais, L.-J. (2004). « La construction de l'identité », dans Denise Deshaies et Diane Vincent (dir.), *Discours et constructions identitaires*, Presses de l'université de Laval.

Gosselin, P., et Le Coguiec, E. (2006). *La recherche création, Pour une compréhension de la recherche en pratique artistique*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18ph3x1>

Guillemette, F., et Luckerhoff, J. (2012). *L'induction en méthodologie de la théorisation enracinée (MTE). Fondements, procédures et usages*, Presses de l'université du Québec. <https://doi.org/10.2307/j.ctv18pgxtm>

Jonnaert, P. (2017). La notion de compétence : une réflexion toujours inachevée, *Éthique publique*, vol. 19 n° 1. <https://doi.org/10.4000/ethiquepublique.2932>

Longuet, F. (2012). *L'impact des outils d'évaluation qualitative et du WEB 2.0 sur le développement et l'identification des compétences professionnelles des enseignants de langues*, thèse de l'université Sorbonne Nouvelle - Paris III. <http://tel.archives-ouvertes.fr/tel-00770640/>

Longuet, F. (2014). Former les enseignants de Français langue étrangère par l'activité de création numérique dialoguée, *Synergies Espagne*, vol. 7, p. 189-204. <http://gerflint.fr/Base/Espagne7/Longuet.pdf>

Longuet, F. (2018). Créativité, dialogisme et multimodalité : trois qualités d'un environnement de conception numérique collaboratif, *Revue internationale de pédagogie de l'enseignement supérieur*, vol. 34, n° 2. <https://doi.org/10.4000/ripes.1450>

Longuet, F., et Springer, C. (2021). *Autour du CECR Volume Complémentaire (2018) : Médiation et collaboration. Une didactique de la relation, écologique et sociosémiotique*, Éditions des archives contemporaines. <https://doi.org/10.17184/eac.9782813004055>

Maturana, H., et Varela, F. (1994). *L'arbre de la connaissance*, Addison Wesley France.

Montiel, H., et Gomez-Zermeño, M. G. (2021). Educational challenges for computational thinking in k-12 education : A systematic literature review of “scratch” as an innovative programming tool, *Computers*, vol. 10, n° 6, p. 69. <https://doi.org/10.3390/computers10060069>

Montiel, H., et Gomez-Zermeño, M. G. (2022). Rock the Boat! Shaken by the COVID-19 Crisis : A Review on Teachers' Competencies in ICT, *Frontiers in Education*, vol. 6, p. 558. <https://doi.org/10.3389/educ.2021.770442>

Moreno-León, J., Robles, G., et Román-González, M. (2015). Dr. Scratch : análisis automático de proyectos Scratch para evaluar y fomentar el pensamiento computacional, *Revista de Educación a Distancia (RED)*, vol. 46, n° 10. <https://doi.org/10.6018/red/46/10>

Morin, E. (2014). *Introduction à la pensée complexe*, Éditions Points.

Okoli, C., et Pawlowski, S. D. (2004). The Delphi method as a research tool : an example, design considerations and applications, *Information et Management*, vol. 42, p. 15-29. <https://doi.org/10.1016/j.im.2003.11.002>

Romero, M. (2016). De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative, *Formation et Profession*, vol. 24, n° 1, p. 87-89. <https://doi.org/10.18162/fp.2016.a92>

Romero, M., Lepage, A., et Lille, B. (2017). Computational thinking development through creative programming in higher education, *International Journal of Educational Technology in Higher Education*, vol. 14, n° 1, p. 1-15. <https://doi.org/10.1186/s41239-017-0080-z>

Rouquette, M.-L. (1973). *La créativité*, Presses universitaires de France.

UNESCO (2018). *Référentiel UNESCO de compétences TIC pour les enseignants*, UNESCO. <https://unesdoc.unesco.org/ark:/48223/pf0000368966>

Veresov, N. (2014). « Émotions, Perezhivanie et développement culturel : le projet inachevé de Lev vygotski », dans C. Moro et N. Muller Mirza (dir.), *Sémiotique, Culture et Développement Psychologique*, Presses universitaires du Septentrion.

Von Uexküll, J. (2010). *Milieu animal et milieu humain*, Bibliothèques Rivages.

Voogt, J., et Roblin, N. P. (2012). A comparative analysis of international frameworks for 21st century competences : Implications for national curriculum policies, *Journal of curriculum studies*, vol. 44, n° 3, p. 299-321. <https://doi.org/10.1080/00220272.2012.668938>

Wilson, A., Hainey, T., et Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts, *Proceedings of the 6th European Conference on Games-Based Learning (ECGBL)*, p. 4-5.

Wing, J. M. (2008). Computational thinking and thinking about computing, *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences*, vol. 366, n° 1881, p. 3717-3725. <https://doi.org/10.1098/rsta.2008.0118>

Wolz, U., Hallberg, C., et Taylor, B. (2011). Scrape : A tool for visualizing the code of scratch programs, *Poster présenté à la 42^e ACM Technical Symposium on Computer Science Education (SIGCSE 2011)*, Dallas, États-Unis.

Pour citer ce chapitre :

Longuet, Frédérique, et Gómez-Zermeño, Marcela Georgina (2024). « Former les enseignants à et par la programmation informatique cocréative en formation initiale », dans Cédric Fluckiger, Laetitia Boulc'h, Sandra Nogry et Christophe Reffay (dir.), *Enseigner, apprendre, former à l'informatique à l'école : regards croisés*, Université Paris Cité, p. 193-210. <https://doi.org/10.53480/2024iecare10x>

