

Mémoire

CAFIPEMF - Enseignement et numérique

L'initiation à la programmation

Comment former les enseignants ?

Julien BARI

CPC - Montbéliard I

2016 - 2017

Table des matières

Introduction.....	1
1. Partie 1 : Cadre théorique	2
1.1. Les Instructions officielles	2
1.2. Le lexique.....	4
1.2.1. Programmation.....	4
1.2.2. Algorithme.....	4
1.2.3. Code.....	5
1.3. La pensée algorithmique	6
1.4. Pourquoi enseigner la pensée algorithmique ?	8
1.5. La démarche scientifique	9
2. Partie 2 : l'ingénierie de la formation.....	10
2.1. Analyse des attentes et des besoins	11
2.1.1. Attentes de l'institution.....	11
2.1.2. Attentes et besoins des enseignants.....	11
2.1.3. Réponse locale aux besoins des enseignants.....	12
2.2. Construction	12
2.2.1. Objectifs.....	12
2.2.2. Dispositif retenu	13
2.2.3. Contenu	14
2.2.4. Démarche	18
2.3. Anticipation	19
2.4. L'accompagnement des enseignants	19
3. Partie 3 : Analyse.....	20
3.1. L'évaluation du module de formation.....	20
3.1.1. A chaud.....	20
3.1.2. Impact sur les pratiques	21
3.1.3. Impact sur les apprentissages des élèves.....	22
3.2. Le suivi en classe.....	23
3.2.1. Module – Partie 1	23

3.2.2.	Module – Partie 2	25
3.3.	Constats et adaptations possibles	25
3.3.1.	Conception, élaboration, anticipation.....	25
3.3.2.	Lors de la mise en œuvre	26
3.4.	Un exemple de productions : Le « Pixel Art »	27
4.	Partie 4 : « Pour aller plus loin ».....	27
4.1.	Formation initiale	27
4.2.	Formation continue – Plan Annuel de Formation.....	28
4.3.	Parcours M@gistère.....	28
4.4.	Partenariats	29
	Conclusion	29

Introduction

L'« initiation à la programmation » dans les nouveaux programmes a suscité de nombreux débats. La première introduction dans le premier degré est lancée en 2014 par Benoit HAMON, alors ministre de l'Éducation Nationale, qui a annoncé favoriser « en primaire une initiation au code informatique, de manière facultative et sur le temps périscolaire ». C'est dans cette optique et en prenant appui sur les recommandations de l'Académie des Sciences¹, qu'a été inscrit le numérique dès septembre 2014, en formation initiale à l'ESPE. L'Académie avait notamment insisté sur l'enseignement de l'informatique en France, et sur le fait qu'« il est urgent de ne plus attendre ». Marianne Thyssen, commissaire européenne en charge de l'Emploi et des Affaires sociales insiste sur le fait qu'actuellement 90% des emplois nécessitent un minimum de compétences en informatique. Outre ce fait qu'il est désormais, et qu'il sera de plus en plus nécessaire d'être à l'aise avec les outils numériques, il est également pointé dans ce rapport un intérêt explicite pour le développement d'une nouvelle forme de pensée appelée la pensée informatique (computational thinking en anglais) : « L'apprentissage de la programmation permet de découvrir les rudiments de la pensée informatique. » M. CHÉNO, Inspecteur Général de l'Éducation Nationale, parle quant à lui de pensée algorithmique.

Un éventail de nouveaux termes (pensée, programmation, code, algorithmes, etc.) est apparu dans les programmes. Cela m'a amené à m'interroger sur les objectifs qui découlaient de cette « initiation à la programmation ». En tant qu'ancien animateur T.I.C.E. et amateur de codage informatique, cela était abstrait pour moi. Je me suis alors questionné sur l'impact que cela pouvait avoir chez certains enseignants. Malgré le fait que quelques-uns d'entre eux comprennent les enjeux de son arrivée dans les programmes, il est indéniable que la plupart ne possèdent pas les connaissances et compétences suffisantes. Les enseignants du second degré étaient 51% en 2016 à faire remarquer que la formation était « inexistante/insuffisante » à la simple utilisation pédagogique des TIC. Concernant le premier degré, les résultats de l'enquête proposée en 2015² ont d'autant plus pointé ce manque de formation, à hauteur de 62%. Le bilan de la concertation sur le numérique à l'école allait également en ce sens, comme le déclarait le Président de la République dans son discours du 7 mai 2015 : « Les conclusions de la consultation sont claires. La première conclusion, c'est qu'il ne peut pas y avoir de diffusion du numérique s'il n'y a pas de formation des enseignants et des personnels. »

Afin de prendre appui sur les représentations initiales, les besoins et les attentes des enseignants du primaire, j'ai diffusé un questionnaire (annexe 1) au niveau national. Celui-ci a été rempli par des enseignants du premier degré, la répartition des cycles étant approximativement équitable (annexe 2). On peut remarquer que la demande de formation, tout comme cela l'était dans l'enquête PROFETIC, est présente : les enseignants sont favorables à 90% environ pour qu'une formation soit proposée sur la programmation (que ce soit sur le plan de circonscription ou sur inscription au niveau du plan annuel de

¹<http://www.academie-sciences.fr/fr/Rapports-ouvrages-avis-et-recommandations-de-l-Academie/l-enseignement-de-l-informatique-en-france-il-est-urgent-de-ne-plus-attendre.html>

² Enquêtes Profétic de 2015 et 2016 - <http://eduscol.education.fr/cid60867/l-enquete-profetic.html>

formation). Une formation sur le sujet paraît alors judicieuse, notamment car la notion de programmation est souvent associée à l'apprentissage d'un langage informatique codé en « Anglais » et où les objectifs d'apprentissages attendus ne sont pas clairs. L'objectif in fine n'est sûrement pas de former des futurs « programmeurs ». Dans quel sens doit-on donc prendre cette « programmation » ? Qu'est-ce que cette « pensée algorithmique » ? Quelles compétences les élèves peuvent-ils développer ? Quels sont les obstacles auxquels cet apprentissage peut-il être confronté ? Peut-on développer la pensée algorithmique sans matériel informatique ? Si oui, comment ?

A partir de ces questionnements, nous allons poser, tout au long de cet écrit, une réflexion autour d'une action de formation, visant à répondre à la problématique suivante : « **Comment amener les enseignants à utiliser la programmation comme un outil au service du développement des compétences permettant la résolution de problèmes au cycle 3 ?** »

Il sera en premier lieu indispensable de clarifier cette « initiation à la programmation » inscrite dans les instructions officielles, de définir les termes qui lui sont associés, que ce soit dans celles-ci ou dans les recherches parallèles. Ensuite, il conviendra d'expliciter les intérêts de l'utilisation de cette programmation, qui est actuellement assez abstraite pour de nombreux enseignants, en définissant les objectifs d'apprentissage, notamment transversaux, les plus importants. Dans un second temps, je m'attacherai à proposer une action de formation à destination d'enseignants, en m'appuyant sur diverses recherches. Enfin, dans la dernière partie, j'analyserai cette formation, qui a pu être mise en place et co-animée avec l'ATICE de la circonscription pour un petit groupe d'enseignants du premier degré ainsi que l'accompagnement réalisé lors de séances en classe.

1. Partie 1 : Cadre théorique

Suite à ce questionnaire personnel, j'ai orienté mes lectures vers diverses sources, me permettant d'éclaircir ce « nouvel » apprentissage, dans un premier temps hors temps scolaire, mais désormais dans les programmes de 2015.

1.1. Les Instructions officielles

Ni la programmation, le code ou même les algorithmes, n'étaient présents, dans les programmes de 2008. Une étude a été réalisée en octobre 2014³ (annexe 3) auprès de 20 ministères de l'Éducation européens afin de savoir si la programmation et le codage faisaient partie des programmes scolaires. C'était le cas pour 12 pays (la Bulgarie, Chypre, la République tchèque, le Danemark, l'Estonie, la Grèce, l'Irlande, l'Italie, la Lituanie, la Pologne, le Portugal et l'Angleterre). Il est à noter que 8 états membres proposent également des cours de codage aux enseignants. Cette étude semble avoir fait écho dans les pays européens, la plupart ayant désormais ajouté, des cours d'informatique et de codage à leur programme scolaire.

³ Computing our future. Computer programming and coding - Priorités, programmes scolaires et actions menées en Europe. European Schoolnet (2014) - <http://www.eun.org/publications/detail?publicationID=481>

C'est ainsi que le Conseil Supérieur des Programmes (CSP) a proposé, dans son projet de programmes des cycles 2, 3 et 4, l'utilisation de logiciel de programmation dès le CE1, tout en préparant le codage bien plus tôt. Cette initiation à la programmation a été renforcée et précisée dans les programmes définitifs du 26 novembre 2015. Cela apparaît explicitement, notamment dans les domaines scientifiques. En mathématiques, un paragraphe est dédié à l'initiation à la programmation dans les repères de progressivité, et en sciences on peut par exemple retrouver « Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles et ludiques ». Au niveau du Socle Commun de Connaissances de Compétences et de Culture, on peut noter des liens avec cette initiation, dans le domaine 1 (les langages pour penser et communiquer) : « Comprendre, s'exprimer en utilisant les langages mathématiques, scientifiques et informatiques » et dans le domaine 2 (Les méthodes et outils pour apprendre) : « En mathématiques, ils apprennent à utiliser des logiciels de calculs et d'initiation à la programmation. ». Un document d'accompagnement du programme intitulé « Initiation à la programmation aux cycles 2 et 3 » a été élaboré, afin d'aider les enseignants dans sa mise en œuvre. Au niveau du collège, un autre document nommé « Algorithmique et programmation », est également disponible pour le cycle 4. Dans cette même optique, le CSP a adopté le projet de programme proposant un enseignement facultatif d'informatique et de création numérique à destination des élèves de première des séries ES, L et S et de terminale des séries ES et L⁴. Dans le domaine 4 du cycle 3, mais également dans bien d'autres endroits, il est intéressant de remarquer qu'un lien étroit est opéré avec la résolution de problèmes : « Les élèves sont graduellement initiés à fréquenter différents types de raisonnement. Les recherches libres (tâtonnements, essais-erreurs) et l'utilisation des outils numériques les forment à la démarche de résolution de problèmes. »

Au niveau du B2i, qu'il s'agisse de la version 2006 ou celle de 2011, aucune allusion à la programmation n'est présente. Le nouveau cadre de référence des compétences numériques qui entrera en vigueur à la rentrée 2017 (annexe 4), intègre, quant à lui, une compétence intitulée « Programmer » (domaine 3 : « Création de contenu »). Un lien est par ailleurs réalisé avec les compétences du Socle Commun de Connaissances de Compétences et de Culture, et avec la pensée algorithmique et informatique, ce référentiel n'étant plus une annexe mais faisant partie intégrante de ce socle. Cela pointe le fait que les compétences associées doivent être intégrées au quotidien et dans tous les domaines. La mise en place de cette initiation va permettre de développer de nombreuses compétences transversales même si toutes ne sont pas explicitées clairement dans les textes. On pourra prendre comme exemple la participation « à des échanges en respectant les règles de communication » ou encore « s'exprimer, argumenter, écouter et donner son point de vue » au niveau du langage oral. Comme le préconise le nouveau programme, la démarche de projet sera à favoriser pour croiser les différents enseignements et permettra de développer le travail en groupe et l'engagement dans un projet collectif entre autres.

⁴<http://www.education.gouv.fr/cid100901/projet-de-programme-pour-un-enseignement-facultatif-d-informatique-et-de-creation-numerique.html>

1.2. Le lexique

1.2.1. Programmation

« On n’imaginera pas un instant que l’on va créer des programmeurs »⁵, c’est ce qu’affirme Laurent CHÉNO, Inspecteur Général de l’Éducation Nationale. Que se cache-t-il alors derrière cette « initiation à la programmation » ? Si nous nous intéressons à l’Histoire, la première machine « programmable » a été élaborée au XVIII^{ème} siècle par Joseph Marie Jacquard : il s’agissait du métier à tisser. Le fait qu’il était possible de la programmer grâce à l’utilisation de cartes perforées, l’amène parfois à être considéré comme l’ancêtre de l’ordinateur ou du robot. La programmation à l’aide de ces cartes permettait de donner l’ordre à la machine de tisser des motifs plus ou moins complexes. Ainsi, programmer était considéré comme le fait de penser à l’avance les instructions que l’on va donner à la machine, afin qu’elle les exécute sans une nouvelle manipulation de la part de l’humain.

Pour autant, la programmation correspond tout de même à l’élaboration d’un programme. C’est vouloir donner une résolution automatique de problème, qu’une machine ou un humain va pouvoir mettre en œuvre seul, d’un point de départ au point d’arrivée. Ainsi, élaborer un programme n’est pas une tâche simple et va nécessiter une organisation et un raisonnement considérables. Colin de la HIGUERA, professeur à l’Université de Nantes et chercheur au laboratoire LINA, indique que la résolution de nombreux problèmes se déroule aujourd’hui en passant par trois étapes⁶, qui rassemble :

- La transformation des données du problème en information (données d’entrée) ;
- Le traitement algorithmique de cette information (recherche d’un algorithme de résolution) ;
- La restitution de la solution sous une forme utile, acceptable, agréable (données de sortie).

La programmation c’est donc le fait d’élaborer un programme, qui est un ensemble d’instructions ou d’algorithmes, transcrit dans un langage prévu pour être lu. Si le destinataire est une machine, on va parler d’un « langage de programmation ». Celui-ci est une expression particulière d’un raisonnement, qui au préalable, a été donné dans un langage compréhensible de tous grâce à un pseudo-langage, que l’on appelle communément une langue.

1.2.2. Algorithme

Le premier algorithme connu a été inventé par Euclide d’Alexandrie au III^{ème} siècle avant J.C., bien célèbre, qui porte aujourd’hui le nom d’ « algorithme d’Euclide » permettant de calculer le PGCD (Plus Grand Commun Diviseur) de deux nombres. Il s’agit d’une suite de divisions euclidiennes, qu’un ordinateur serait capable de réaliser s’il était programmé pour cela. Pourtant, le terme d’ « algorithme » n’apparaît qu’au IX^{ème} siècle, prenant son origine du nom latinisé de Muhammad ibn Musa AL-KHWARIZMI. Il est généralement écrit dans un langage plus proche de l’Homme que de la machine.

⁵ Lycée Jean Zay, Paris, vendredi 4 décembre 2015 « Algorithmique au cycle 3 »

⁶ <http://www.slate.fr/story/110897/ecole-enseigner-informatique-coding>

Voici une définition, proposée en 1936 par Alan TURING : « Un algorithme est une suite finie et non-ambiguë d'instructions permettant de donner la réponse à un problème. » De manière simplifiée, nous pouvons reprendre la définition donnée par Gérard Berry, professeur, chaire Algorithmes, machines et langages, Collège de France, membre de l'Académie des sciences et de l'Académie des technologies⁷ : « Les algorithmes sont des procédés de calculs automatiques sur l'information. »

Donald Ervin Knuth⁸, quant à lui, liste cinq propriétés, comme étant les prérequis d'un algorithme :

- La finitude (un algorithme doit toujours se terminer avec un résultat) ;
- Une définition précise (chaque étape doit être explicite, sans ambiguïté) ;
- Des entrées (quantités qui lui sont données avant qu'il ne commence) ;
- Des sorties (quantités ayant des relations avec les entrées) ;
- Un rendement (les opérations doivent être suffisamment basiques pour pouvoir être réalisées par un Homme dans une durée finie).

Dans le programme, ce terme d'algorithme n'est pas présent uniquement dans une partie liée à une initiation à la « programmation », nous pouvons le retrouver en cycle 2 dans le sous-domaine des « Nombres et Calculs » : « Mettre en œuvre un algorithme de calcul posé pour l'addition, la soustraction, la multiplication. » En effet, les étapes pour trouver le résultat d'un calcul posé sont toujours les mêmes, c'est donc grâce aux algorithmes que les premiers calculs automatisés ont pu être créés à l'aide de machines, dans un premier temps, mécaniques. Ainsi, un algorithme est une suite finie et non-ambiguë d'instructions rédigée dans un pseudo-langage, comme le Français, et qui peut être traduit dans un langage de programmation quelconque.

1.2.3. Code

Le code est, d'une manière générale, un langage. Dans le domaine informatique, il s'agit d'un langage de programmation (il en existe de nombreux) qui représente l'information. Comme le précise Alexander REPENNING « chaque langage se distingue par sa syntaxe qu'on doit maîtriser pour rendre le programme fonctionnel⁹ ». Il est devenu, en France, un sujet important de débat en février 2013 lorsque Barack OBAMA participait à l'annonce pour la campagne « Hour of Code » (semaine dédiée à l'éducation informatique)¹⁰ et incitait les jeunes à devenir acteur plutôt que consommateur, en créant des jeux vidéo. La représentation de ce « code » informatique est généralement perçue comme une suite de requêtes et d'instructions exécutables, écrites en anglais à l'aide d'un vocabulaire spécifique (if, else, while, for, etc.). Ainsi, la programmation est ressentie comme pénible et ennuyeuse, car l'encodage est long, compliqué, et

⁷ http://www.dailymotion.com/video/x1zvso5_la-pensee-informatique-coeur-du-monde-numerique-gerard-berry_school - La pensée informatique, cœur du monde numérique : Gérard BERRY

⁸ Il est un des pionniers de l'algorithmique et a fait de nombreuses contributions dans plusieurs branches de l'informatique théorique.

⁹ « La pensée informatique dans la formation des enseignants » - Alexander REPENNING

¹⁰ <https://www.youtube.com/watch?v=6XvmhE1J9PY> - the Hour of Code campaign for Computer Science Education Week 2013

également en raison du fait que l'on ne reconnaît pas l'idée dans le code. Des élèves de cycle 3, voire de cycle 2 sont-ils réellement capables de « coder » dans ce langage ? Il est évident que ce n'est pas le cas, et cet enseignement n'a pas pour but de faire des élèves de futurs programmeurs, comme cela a déjà été souligné. Dans le document d'accompagnement à l'initiation à la programmation aux cycles 2 et 3, nous pouvons retrouver : « Aux cycles 2 et 3, les ambitions sont assez modestes : il s'agit de savoir coder ou décoder pour prévoir ou représenter des déplacements, de programmer les déplacements d'un robot ou ceux d'un personnage sur un écran. »

L'utilisation d'un code très simple sera donc privilégiée, car la liste des instructions possibles reste très réduite pour des déplacements. Ainsi, de simples flèches peuvent représenter une instruction d'orientation ou de mouvement. Il est à noter que ce code doit être connu de tous et être explicite, à l'instar d'une légende sur une carte, car une flèche dirigée vers la droite pourrait signifier « se tourner vers la droite », « avancer vers la droite », « avancer dans la direction actuelle » ou encore « se tourner et avancer vers la droite ». A l'aide d'un code composé de 4 flèches, chacune signifiant « se tourner et avancer d'une case en direction de... », il est alors totalement possible pour des élèves de cycle 1 de diriger un personnage, un robot ou même une personne.

D'après la formation « Class'Code »¹¹, « Coder est l'activité qui permet de voir se concrétiser ses idées. Mais rien n'empêche ces idées de naître et de mûrir en dehors de toute forme de programmation (et donc même sans ordinateur !), en utilisant simplement le raisonnement comme moteur principal. » Le code correspond donc à la finition qui pourra être lue, et non à la réflexion de construction, qui correspond plus à la pensée algorithmique, que nous allons développer ci-après.

1.3. La pensée algorithmique

Si l'on reprend la procédure proposée par Colin de la HIGUERA permettant de résoudre un problème, en partant de la transformation en information vers une résolution informatique, cela s'appelle, en anglais, le « computational thinking ». Généralement, nous rencontrons plusieurs traductions possibles, dont les principales sont la pensée algorithmique, la pensée informatique, la pensée computationnelle et la pensée logique. Laurent CHÉNO, Inspecteur Général de l'Éducation Nationale insiste sur le fait qu'il s'agisse surtout « d'initiation à la pensée algorithmique, plus qu'à la programmation ». Il développe cette pensée algorithmique et liste quelques compétences liées, ainsi que d'autres plus générales, dont toutes peuvent être travaillées dans d'autres champs que les mathématiques :

- Savoir décomposer un problème en tâches simples ;
- Savoir reconnaître des tâches que l'on a déjà effectuées, ou qui se répètent (réutilisabilité de codes, répétition de schéma) ;

¹¹ <https://pixees.fr/classcode/accueil/> - Class'Code est un « MOOC » (massive open online course), une formation gratuite à distance à destination de toutes personnes désireuses d'initier les jeunes à la pensée informatique.

- Apprendre à travailler ensemble à un projet commun (cette pensée favorise l'autonomie, mais également le travail collaboratif) ;
- Favoriser l'imagination, le sens créatif.

Il met également en avant les activités dites « débranchées », en insistant sur le fait que l'on peut avoir une vision algorithmique en définissant un jeu, ses règles et en les appliquant. Cette vision est par exemple très importante dans les activités physiques et sportives, et permet d'imaginer, d'élaborer et d'exécuter des séquences répondant à diverses finalités (de défense, d'attaque, de progression, de possession, etc.).

Dans le document d'accompagnement intitulé « Initiation à la programmation aux cycles 2 et 3 », cette pensée algorithmique n'est pas développée, mais elle apparaît explicitement dans le nouveau cadre de référence des compétences numériques et dans le programme de cycle 4. Pierre Tchounikine, professeur à l'Université Grenoble-Alpes, s'est intéressé plus en détail à l'intégration de cette programmation dans les programmes et a proposé au préalable d'initier les élèves à la « pensée informatique ». Ce professeur précise que celle-ci doit être, comme dans tout apprentissage, acquise par les enseignants. Il explicite qu'elle implique « d'appréhender le monde selon l'approche employée en programmation par les développeurs de logiciels » et de ce fait propose de scinder cette approche en cinq catégories, comme selon les promoteurs du logiciel Scratch :

- Abstraction (appréhender un problème et sa solution à différents niveaux) ;
- Algorithmes (réfléchir aux tâches à accomplir sous forme d'une série d'étapes) ;
- Décomposition (comprendre que pour résoudre un problème complexe il faut le décomposer en plusieurs problèmes simples) ;
- Reconnaissance de formes (comprendre qu'il est probable qu'un nouveau problème soit lié à d'autres problèmes déjà résolus par l'élève) ;
- Généralisation (réaliser que la solution à un problème peut servir à résoudre des problèmes semblables).

Quelle est alors la différence entre pensée informatique et pensée algorithmique ? Si nous croisons ces compétences avec celles proposées par M. CHÉNO, nous pouvons dire que la pensée algorithmique est une partie de la pensée informatique, qui correspond au raisonnement à mettre en œuvre pour résoudre le problème. La pensée informatique englobera le traitement des informations en entrée, la pensée algorithmique, l'encodage dans un langage choisi et la diffusion en sortie. Ainsi, même si l'accent est mis sur la pensée algorithmique dans le nouveau programme, la pensée informatique est tout de même travaillée lors de toute résolution de problème, dans tous les domaines.

Ce terme de « pensée informatique » est nettement plus employé depuis la parution d'un article de la chercheuse en informatique Américaine Jeanette Wing, qui confirme notre réflexion, en la définissant ainsi : « La pensée informatique est un processus de réflexion qui visualise tant la formulation d'un problème que la représentation de sa solution de façon à ce qu'elles puissent être exécutées par des humains ou des machines. ». La formation Class'Code propose une définition très proche : « C'est la capacité, par rapport à quelque chose que l'on fait ou que l'on sait faire, que l'on veut faire faire par une

machine, à décomposer cette chose en tâches simples, en instructions non ambiguës, que l'on va pouvoir faire exécuter par la machine à notre place. C'est une façon de réfléchir qui permet de formaliser un processus qui peut être inconscient et de le rendre exécutable par d'autres. »

Enfin, tout comme M. CHÉNO l'indique pour la pensée algorithmique, Colin de la HIGUERA explicite le fait que la pensée informatique peut également s'apprendre à l'aide d'activités débranchées : « la pensée informatique n'a même pas besoin d'un ordinateur pour avoir un sens ! La démarche procédurale, l'abstraction, la capacité de créer des algorithmes sont autant de qualités à apprendre puis à maîtriser pour résoudre des problèmes de toutes natures. Le code nous permet d'exécuter cette résolution. L'informatique nous permet de la comprendre. C'est cette matière qu'il convient d'enseigner. »

1.4. Pourquoi enseigner la pensée algorithmique ?

L'enseignement de cette pensée algorithmique ne doit pas avoir de finalité professionnelle, comme le soulignait M. CHÉNO. Elle doit révéler des vocations et permettre aux élèves de se destiner aux études de leurs choix. Les objectifs informatiques, au sens des compétences liées à l'utilisation d'un ordinateur, restent donc secondaires. Tout au long de leur vie, les élèves seront confrontés à la réalité et à de nombreux problèmes du quotidien, que ce soit au niveau personnel ou au niveau professionnel. Pour résoudre certains de ces problèmes, notamment les plus complexes, qu'ils soient généraux (comme la gestion d'un budget) ou spécifiques (comme l'organisation de la production dans une boulangerie), ils devront mettre en œuvre de nombreuses compétences. Mitch Resnick, directeur du groupe « Lifelong Kindergarten » (que l'on pourrait traduire par « À la maternelle toute la vie ») au MIT Media Lab, listait quelques compétences, lors d'une conférence¹² TEDx filmée en novembre 2012, que les élèves développent lorsqu'ils programment, qui ne sont pas forcément liées à l'ordinateur et qui leur seront utiles durant leur avenir :

- Tester de nouvelles idées ;
- Prendre des idées complexes et les décomposer en étapes simples ;
- Collaborer avec d'autres personnes sur un sujet ;
- Identifier et résoudre des bugs, quand il y a des erreurs ;
- Rester concentré, persévérer face à la frustration ;

Il insiste sur le fait que peu importe l'avenir professionnel des élèves, ces compétences sont utiles et il réalise une analogie pertinente avec le langage : lorsque nous apprenons à lire et à écrire, ce n'est pas forcément pour devenir écrivain professionnel. Ainsi, ces compétences de pensées créatives, de raisonnement systématique, de travail collaboratif entre autres, peuvent être développées en programmant. Il s'agira d'apprendre à programmer, mais également de programmer pour apprendre.

Les compétences informatiques liées à l'ordinateur ne sont pas une priorité, mais ne sont pas pour autant, à mettre de côté. Günther Oettinger, commissaire européen à l'Économie numérique a déclaré :

¹² https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code?language=fr#t-799323

« Dans cinq ans, 825 000 emplois seront vacants, tout simplement parce que les employeurs ne trouveront pas de travailleurs avec les bonnes compétences en informatique. » Il convient toutefois de prendre du recul et de ne pas former les enseignants au « tout informatique », l'enquête PISA « Connectés pour apprendre¹³ », parue en 2015, démontrant que les performances en compréhension de l'écrit électronique et la qualité de la navigation chutent dès lors que l'on utilise l'outil numérique pour le travail scolaire plus d'une fois par semaine.

1.5. La démarche scientifique

La pensée algorithmique permet donc de proposer une solution à un problème, qui pourrait, avec un très bon algorithme, résoudre de nombreux problèmes similaires. Cependant, avant de pouvoir imaginer résoudre « automatiquement » des problèmes, il conviendra d'amener les élèves à apprendre comment résoudre un problème, qu'il soit dans un domaine scientifique ou dans un tout autre domaine. Dans le nouveau programme de cycle 3, en « sciences et technologie », nous pouvons remarquer que la première partie intitulée « Pratiquer des démarches scientifiques et technologiques », précise que l'on doit amener les élèves à utiliser ce type de démarche pour résoudre un problème, notamment lié à une question de nature scientifique et technologique. Le document d'accompagnement le confirme, en mettant en avant que l'initiation à la programmation permet de travailler « l'adoption d'une démarche scientifique ». En effet, la réalisation d'un programme, qu'il soit simple ou complexe, engage l'élève dans une résolution de problème : « Comment vais-je m'organiser ? Par quoi dois-je commencer ? Comment pourrais-je vérifier ? ». Il s'agira là d'accompagner les élèves dans la pratique systématique d'une démarche scientifique pour résoudre un problème donné.

Qu'est-ce que la démarche scientifique ? C'est un ensemble d'étapes, nécessaire pour essayer de répondre aux questions que l'on se pose. Dans le volet 2 du cycle 3, on peut retrouver : « Les élèves sont graduellement initiés à fréquenter différents types de raisonnement. Les recherches libres (tâtonnements, essais-erreurs) et l'utilisation des outils numériques les forment à la démarche de résolution de problèmes. ». Nous allons donc les inviter à utiliser le code comme un outil, permettant de résoudre, dans un premier temps, des problèmes simples, afin de développer leur raisonnement.

Allons plus loin et cherchons quelles sont les étapes de cette démarche d'investigation, dite scientifique. Dans le nouveau programme de cycle 3, ces étapes sont déclinées ainsi :

- Formuler une question ou une problématique scientifique ou technologique simple ;
- Proposer une ou des hypothèses pour répondre à une question ou un problème ;
- Proposer des expériences simples pour tester une hypothèse ;
- Interpréter un résultat, en tirer une conclusion ;
- Formaliser une partie de sa recherche sous une forme écrite ou orale.

¹³ <https://www.oecd.org/fr/edu/scolaire/Connectes-pour-apprendre-les-eleves-et-les-nouvelles-technologies-principaux-resultats.pdf>

La troisième phase est une des caractéristiques particulières de la démarche scientifique : les élèves seront invités à expérimenter leur code, avec ou sans outil numérique. C'est ici que nous engagerons les élèves à faire des recherches libres, à tâtonner et à faire des « essais-erreurs » : le statut de l'erreur aura un rôle très important à jouer durant la mise en œuvre du projet.

L'objectif principal de l'initiation à la programmation est donc de rendre les futurs citoyens autonomes, en développant leur raisonnement et leur créativité afin de résoudre des problèmes de plus en plus complexes. La programmation est utilisée ici comme un outil amenant particulièrement les élèves à pratiquer des démarches scientifiques, puis en les aidant à développer une pensée algorithmique permettant des résolutions plus efficaces. C'est sur cela qu'il faudra insister lors de la formation des enseignants, mais également sur le fait que ces compétences peuvent être développées à l'aide d'activités débranchées et sur le rôle important du langage oral. Un autre point important sera de fixer le lexique que nous venons de développer.

2. Partie 2 : l'ingénierie de la formation

En France, des recherches sur les intérêts d'un apprentissage de l'informatique et sur la formation des enseignants ont émergé. Par exemple l'actuel projet « DALIE » (Didactique et Apprentissage de L'Informatique à l'Ecole) « se donne comme objectif de produire un livre blanc sur la place et la nature d'un enseignement à l'informatique à l'école primaire et ses conséquences sur la formation des enseignants¹⁴ ». Ce projet prend également appui sur la prise de conscience liée au fait que « l'algorithmique et la programmation, comme savoir faire-faire, sous leurs différentes formes, sont au cœur de l'informatique. » Celui-ci étant très récent, les premiers résultats de ce projet ne sont pas encore disponibles. Cependant, de très nombreuses formations sont tout de même proposées.

Les enseignants sont peu nombreux à mettre en place des séances liées à la programmation. Les résultats de mon questionnaire indiquent que 21% seulement effectuent des séances sur cette thématique. Cela est dû à plusieurs raisons, dont la principale est liée à un manque de formation. Ainsi, je me suis attaché à concevoir et à élaborer une formation tentant de répondre à la problématique posée. Cette dernière a été conçue en prenant appui sur le document proposé par l'Ifé (Institut français de l'éducation), intitulé « Concevoir des formations pour aider les enseignants à faire réussir tous les élèves¹⁵ » paru en septembre 2016. L'équipe du centre Alain-Savary explicite le fait que toute action de formation se déroule en deux phases : la conception et sa mise en œuvre, cette première se déclinant en trois temps, à savoir l'analyse, la construction et l'anticipation.

¹⁴ <http://www.unilim.fr/dalie/le-projet/>

¹⁵ <http://centre-alain-savary.ens-lyon.fr/CAS/documents/publications/concevoir-des-formations>

2.1. Analyse des attentes et des besoins

La première partie consiste à clarifier plusieurs points : la commande institutionnelle, les moyens, le négociable et le non-négociable.

2.1.1. Attentes de l'institution

Contrairement au cycle 4, le programme de l'élémentaire ne mentionne que quelques liens avec l'outil informatique, notamment concernant l'utilisation d'un logiciel de programmation adapté et de logiciels de géométrie dynamique. Pour le primaire, si nous faisons correspondre le document d'accompagnement avec le programme, on remarque que les attentes de l'institution sont surtout liées au fait d'amener les élèves à :

- Adopter une démarche scientifique ;
- Développer l'abstraction afin d'anticiper.

L'objectif est donc de rendre les élèves autonomes dans une résolution de problème afin qu'ils usent de démarche scientifique de manière efficiente, c'est-à-dire en réduisant les essais-erreurs. Pour cela, le développement de leur anticipation sera important et sera bien plus efficient si les élèves développent ce que l'on a appelé la « pensée algorithmique » en première partie. Ces deux aspects seront donc importants à traiter durant le module de formation, à savoir « la démarche scientifique » et « la pensée algorithmique ».

2.1.2. Attentes et besoins des enseignants

Les résultats du questionnaire (annexe 2) dévoilent que les enseignants sondés pensent que l'utilisation d'outils numériques (ordinateurs et/ou tablettes) est indispensable, à hauteur de 58% d'entre eux. Pourtant, comme nous l'avons déjà remarqué, les compétences attendues pour cette initiation sont bien, pour la plupart, transversales, et non vers un but de formation d'outils informatiques. Comme l'indique M. CHENO, nous pouvons développer cette pensée sans outils numériques. Parmi tous ces enseignants, 13% ont déjà eu l'occasion de suivre une formation sur le sujet. Il est important de remarquer que, dès le début, parmi les 137 réponses, seulement 19% des enseignants pensent que sa mise en place est facile, voire aisée, tandis que près de 55% l'imaginent difficile, voire très difficile. On observe que seulement 14% ont déjà suivi une formation sur le sujet, et parmi ceux-ci, le choix fut volontaire à 63% (Inscription au plan annuel de formation ou formation hors Education Nationale).

Quels sont alors leurs attentes et leurs besoins ? Bien que 7% d'entre eux mettent une priorité sur l'appropriation du lexique, on remarque que seuls 8% des enseignants indiquent être capable de différencier et définir clairement les termes « algorithme », « code » et « programme ». Ainsi, l'appropriation de ces termes sera donc une partie incontournable de cette formation pour que le langage utilisé avec les élèves soit adéquat. Par ailleurs, 75% des enseignants n'ont aucune référence théorique liée à cet apprentissage, et cela est explicité lorsqu'on leur demande leurs attentes si une formation leur serait

proposée : ils sont un peu plus de 47% à souhaiter des outils concrets et environ 37% à vouloir être formés sur les logiciels de programmation et/ou les robots. Enfin, les dernières attentes concernent la compréhension des enjeux, de ce qui doit être réellement enseigné et l'organisation pédagogique des séances. A noter que quelques-uns ont émis le souhait d'avoir un accompagnement en classe, qui ferait suite à ladite formation.

2.1.3. Réponse locale aux besoins des enseignants

D'un point de vue plus pratique, six enseignants volontaires de la circonscription ont formulé le souhait d'avoir une formation sur le sujet. Cette dernière a été dispensée et une analyse sera présentée ultérieurement dans ce document. Deux autres enseignants d'une autre circonscription, également intéressés par le sujet, ont souhaité rejoindre ce groupe et assister au module de formation, tous les cycles de l'école primaire étant représentés. Afin d'agir et de répondre au plus près de leurs attentes et de leurs besoins, les réponses du questionnaire données par ces enseignants ont été traitées indépendamment. Concernant leurs attentes, ceux-ci ont indiqué souhaiter être principalement formés aux outils, que ce soit sur le logiciel Scratch ou sur un robot. Il est à noter que ce groupe est composé d'enseignants volontaires, mais avec un niveau hétérogène de maîtrise des outils numériques. Ils sont également 7 sur 8 à indiquer qu'un outil (ordinateur et/ou tablette) est nécessaire pour la mise en place de cet apprentissage. L'étude des activités débranchées sera donc incontournable. Enfin, bien que la plupart indiquent que la mise en œuvre de séances liées à la programmation paraisse aisée ou facile, 6 hésitent quant à la définition du lexique associé.

Plusieurs d'entre eux avaient déjà mis en place des activités liées à la programmation. Cependant, ce besoin de formation s'est fait ressentir lors des premiers échanges, compte-tenu du fait que tous avaient commencé cet apprentissage sans en voir les intérêts et les objectifs principaux. En effet, les séances se déroulaient directement sur le logiciel Scratch sans qu'une compétence précise soit ciblée. Lors de ces échanges, il a été remarqué que, malgré le fait qu'ils expliquaient que les élèves développaient des compétences, celles-ci étaient toutes liées à l'outil. Il a été également intéressant d'évoquer la place grandissante du langage oral au sein de leurs séances, qui était presque inexistant au départ. Ainsi, le contenu envisagé précédemment, lié à la pensée algorithmique, aux démarches scientifiques et au rôle important du langage oral prenait tout son sens.

2.2. Construction

2.2.1. Objectifs

Une fois les attentes et les besoins clarifiés, il convient de définir les objectifs de ce module de formation. Il est pertinent, dans un premier temps, d'amener les enseignants à prendre conscience qu'il est nécessaire, pour cet apprentissage, que les élèves développent les compétences liées à la résolution de problèmes et notamment qu'ils s'habituent à entrer dans des démarches scientifiques. Des infographies explicitant les étapes de ce type de démarche peuvent être utilisées, un support de présentation doit donc être élaboré et la salle de formation doit pouvoir permettre de le projeter. Par ailleurs, afin d'utiliser le

vocabulaire adapté, il est nécessaire de clarifier les premiers termes, à savoir « programmation / programme », « algorithme », « coder », « informatique » et « langage ». Enfin, dans le but d'accompagner les enseignants dans le démarrage de cet apprentissage, une séquence sera proposée, permettant de valoriser les intérêts de l'utilisation d'activités débranchées et de la démarche de projets.

Dans un second temps, il est intéressant que les enseignants soient formés à la « pensée algorithmique », ce qui est, in fine, l'objectif d'apprentissage, qui est par ailleurs davantage explicité et développé dans les textes institutionnels du cycle 4. Il convient de les mettre dans des situations progressives de résolution de problèmes, qui les amènent à s'approprier ce concept, notamment l'utilisation successive de démarches scientifiques, permettant une reconnaissance de schéma et une anticipation croissante, amenant de fait, à des résolutions de plus en plus efficaces.

Nous pourrions énoncer les objectifs des formés ainsi :

- Comprendre les objectifs d'apprentissage transdisciplinaires, et notamment les intérêts de l'utilisation de démarches scientifiques dans la résolution de problème ;
- Clarifier et différencier le lexique (notamment « programmation / programme », « algorithme », « coder », « informatique » et « langage ») ;
- Mesurer et donner les principaux intérêts de l'utilisation d'activités débranchées, de la démarche de projet et du rôle important du langage oral ;
- S'approprier une séquence d'apprentissage ;
- Comprendre, expliquer et développer une « pensée algorithmique » ;
- Maîtriser un logiciel de programmation adapté aux élèves.

2.2.2. Dispositif retenu

Afin d'enrôler les enseignants, même si ceux-ci sont volontaires, il est primordial de proposer une formation en présentiel, même si la mise à disposition des documents sur un parcours en ligne type « M@gistère » peut être envisagée par la suite. Cette formation pourrait s'inscrire au sein :

1. Du plan annuel de formation départemental ou académique ;
2. Du plan annuel de formation continue proposée au sein de la circonscription ;
3. D'un conseil de cycle d'une même école ;
4. D'une formation sur la base du volontariat regroupant plusieurs écoles.

Malheureusement, cela est impossible à mettre en place dans les deux premiers cas. En effet, concernant le plan de formation au niveau départemental, les propositions sont soumises et validées par la DSDEN bien avant la rentrée scolaire. Le plan de formation de circonscription, quant à lui, avait également déjà été acté et ne comprenait pas d'action de formation sur ce thème. Toutefois, il sera envisagé dans ce document de proposer une formation sur l'un de ces deux formats. La troisième possibilité, à savoir une formation au sein d'une seule école serait intéressante à envisager, afin de pouvoir analyser en équipe la progressivité

des apprentissages. Cependant, cela ne me permet pas d'avoir une analyse poussée, le cadre serait alors homogène (matériel, organisation, etc.). Les demandes de formation et/ou d'accompagnement reçues à ce sujet sur tous les cycles et émanant de diverses écoles, ont amené Mme l'Inspectrice de l'Éducation Nationale à proposer un module de formation sur le sujet, composé d'enseignants de tous niveaux (débutants ou expérimentés). Le dernier format a donc été retenu. Il est à noter que, compte-tenu du sujet et des modalités d'expérimentation, l'organisation de formations à un groupe restreint de stagiaires a été privilégié afin de maximiser les échanges, que ce soit avec le(s) formateur(s) ou entre participants, qui plus est, pour une formation liée aux TICE où l'utilisation des outils numériques lors de la formation est à favoriser.

Les objectifs de ce module de formation ne peuvent être atteints en seulement deux ou trois heures, nous opterons donc pour un module de quatre heures. Pour éviter tout risque lié au fait que les enseignants sortent de la formation en se sentant « délaissés » (sans et/ou d'objectifs précis), il a été choisi de mettre en place deux formations de deux heures, co-animées avec l'ATICE de la circonscription qui pourrait accompagner les enseignants dans les classes. Elles seront espacées d'une période complète, permettant ainsi de mettre en place la première partie de la séquence (activités débranchées) et d'en faire un retour lors de la seconde formation.

2.2.3. Contenu

1^{er} présentiel

Le contenu de la formation étant dense, il faut réfléchir à une répartition pertinente des objectifs. Ainsi, il est judicieux de travailler, lors de la première formation, sur les bases, à savoir :

- Comprendre les objectifs d'apprentissage transdisciplinaires, et notamment les intérêts de l'utilisation de démarches scientifiques dans la résolution de problème

D'après les résultats du questionnaire (questions 9 et 13), on peut remarquer que les enseignants comprennent, d'une manière générale, les bénéfices que cet apprentissage peut apporter. Malgré tout, il est important d'insister sur ces intérêts, afin de s'assurer de la justesse de ceux-ci. En effet, les données indiquent tout de même qu'ils sont généralement incertains : bien que la plupart émettent un avis positif, ils sont, pour chaque proposition, environ 66% en moyenne à être « Plutôt d'accord » ou « Plutôt pas d'accord ». Ainsi, il est judicieux de sélectionner et d'analyser (sur un support de présentation¹⁶) les parties du socle commun (domaines 1, 4 et 5), des programmes (tous cycles) et bien entendu, du document d'accompagnement proposé pour les cycles 2 et 3. Dans ce dernier document, nous pouvons pointer le paragraphe présentant les intérêts des activités débranchées, et les trois compétences principales. Cela permet d'aborder les démarches de résolution de problème, à savoir « adopter une démarche scientifique », puis la pensée algorithmique (« développer l'abstraction : apprendre à anticiper l'effet de

¹⁶ Support de type « PREZI » qui a servi de support à la première formation : <https://prezi.com/wflpoizcxgil/>

telle ou telle séquence d'instructions avant même de la faire exécuter »). Nous pouvons alors proposer aux formés d'étudier la démarche scientifique dans un premier temps, l'analyse de la pensée algorithmique, c'est-à-dire l'anticipation, l'abstraction, la reconnaissance de schémas, etc. sera réalisée lors de la seconde partie du module. Afin d'être plus précis dans les attentes, une étude des repères de progressivité serait opportune.

- Clarifier et différencier le lexique (notamment « programmation / programme », « algorithme », « coder », « informatique » et « langage »)

Les différents termes abordés ci-dessus doivent être explicités, à l'aide de définitions simples et dans un ordre logique. J'ai ainsi pensé qu'il serait intéressant de commencer par définir la notion de programmation comme « l'ensemble des activités qui permettent l'écriture des programmes informatiques » (Cf. : Wikipédia). Nous pourrions alors échanger autour du terme « informatique », qui est généralement lié à l'outil et non à la « Science du traitement automatique et rationnel de l'information considérée comme le support des connaissances et des communications », comme cela est défini simplement par le Larousse. Le terme coder, quant à lui, pourra être explicité par une phrase tirée du site de la DANE de l'académie de Besançon : « « Coder » est le fait d'écrire une suite d'instructions dont l'ensemble constitue un programme. », ce qui permettra d'aborder le terme de programme. Ce dernier pourrait être défini comme étant « un ensemble d'instructions et de données représentant un algorithme et susceptible d'être exécuté par un ordinateur. » (Cf. : Larousse). Enfin, grâce à ce cheminement, nous pourrions aborder la notion d'algorithme, qui pourra être définie comme l'a proposée Alan TURING en 1936 : « Un algorithme est une suite finie et non-ambiguë d'instructions permettant de donner la réponse à un problème. »

- Mesurer et donner les principaux intérêts de l'utilisation d'activités débranchées, de la démarche de projet et du rôle important du langage oral

Lors de l'étude du document d'accompagnement, nous aurons abordé le rôle important des activités débranchées (qui ne nécessitent aucun outil numérique). Il est intéressant de développer cet aspect en insistant sur les deux principales raisons, à savoir l'intérêt de se focaliser sur l'apprentissage des compétences transversales en évitant d'ajouter les difficultés liées à la manipulation des outils numériques et ainsi éviter de penser en compétences informatiques. En effet, la résolution de problèmes grâce à la pratique des démarches scientifiques peut s'opérer dans de nombreux contextes sans matériel numérique. Par ailleurs, il sera opportun d'échanger à propos de la démarche de projet, mis en avant dans les nouveaux programmes et dans le socle commun (domaine 2) : « Coopérer et réaliser des projets convoquent tous les enseignements. La démarche de projet développe la capacité à collaborer, à coopérer avec le groupe en utilisant des outils divers pour aboutir à une production. ». Enfin, à travers tous les échanges du module de formation, il sera important de pointer les intérêts du travail par groupes et le rôle important du langage (oral et écrit) permettant de mettre en mots le raisonnement et l'organisation de chacun.

- S'approprier une séquence d'apprentissage

Le présent mémoire n'est pas un mémoire de recherche, et de ce fait, je me suis appuyé sur des séquences reconnues. Afin d'être au plus près des attentes institutionnelles, les séquences proposées sont celles issues du projet « 1, 2, 3 codez » élaboré par « La main à la pâte » et soutenu par le ministère. Celles-ci marquent explicitement l'intérêt du passage par les activités débranchées, la première partie de chaque cycle ne nécessitant aucun outil numérique, le travail étant axé sur le raisonnement et l'apprentissage de la démarche scientifique. Nous présenterons au préalable une situation de référence (un chat essaie de sortir d'un labyrinthe, annexe 5) pour apprécier les conceptions initiales des élèves au niveau du code. Vont-ils produire un texte (ou le dicter à l'adulte pour les élèves de cycle 1) ? Leur proposition comportera-t-elle un autre langage, peut-être codé (comme de simples flèches) ? Sera-t-elle constituée de boucles ? Cette situation pourra servir de point d'appui pour une évaluation de l'impact de la formation sur les apprentissages des élèves. Enfin, des formations et divers outils que les enseignants pourront découvrir seuls si besoin seront présentés (Class'Code, site 1, 2, 3 codez, CS Unplugged, code.org et Scratch).

2nd présentiel

Concernant la seconde partie de la formation, nous réinvestirons la démarche scientifique pour amener les enseignants à remarquer que les essais-erreurs permettent, si on les garde en mémoire, d'éviter de reproduire les mêmes erreurs. Ceci aura pour but d'amener les élèves à anticiper les réponses, afin de rendre leurs démarches plus efficaces. On parlera là, après un retour sur la mise en place de la première séquence composée d'activités débranchées, de la pensée algorithmique¹⁷.

- Comprendre, expliquer et développer une « pensée algorithmique »

Dans un premier temps, avant d'aborder la notion de pensée algorithmique, il est pertinent de s'assurer que l'apprentissage vers un usage des démarches scientifiques a bien été réalisé, notamment lors des retours sur les séances. Puis, il sera intéressant d'avoir un temps d'échanges avec les formés, afin de les amener à définir approximativement la pensée algorithmique comme étant « Le raisonnement à mettre en œuvre pour résoudre un problème de manière répétitive à l'aide d'un algorithme, le plus efficace possible. » Cette façon de penser se déclinant en trois principales étapes :

- Savoir décomposer un problème en tâches simples ;
- Savoir reconnaître des tâches déjà effectuées, ou qui se répètent (répétition de schéma, réutilisabilité de codes) ;
- Anticiper des tâches qui se ressembleraient ;

¹⁷ Support de type « PREZI » qui a servi de support à la seconde formation : <https://prezi.com/wvrmevdkh58p/>

Cette façon de penser n'est pas nouvelle : nous essayons depuis toujours, d'amener les élèves à réinvestir au maximum ce qu'ils ont appris, dans d'autres contextes. Or, pour ce faire, il est nécessaire que notre enseignement ne soit pas trop cloisonné, ces apprentissages ayant pour objectif de former les futurs citoyens. Ainsi, afin que cela ait du sens et que les enseignants comprennent davantage les intérêts, il sera pertinent de leur présenter des situations concrètes. Nous pouvons prendre comme exemple l'organisation des tâches, synchrones et asynchrones, d'un boulanger, pour que le pain soit prêt et chaud en fonction des heures d'affluence. Quelles sont ses heures d'affluence ? A quelle heure va-t-il allumer le four ? Combien de temps met-il pour pétrir sa pâte ? Quelle quantité de pain va-t-il préparer ? Tant de questionnements qui en font une situation complexe, dont l'organisation nécessitera une pensée algorithmique poussée et dont les essais-erreurs auront un rôle important à jouer. Il en va de même pour l'électricien. A quel moment doit-il commander son matériel, et pour quel chantier ? Comment va-t-il gérer ses rendez-vous, son emploi du temps, et anticiper certains imprévus qui se sont déjà produits ? Bien entendu, en primaire, les élèves ne sont certainement pas capables d'atteindre un niveau de pensée aussi abstrait. Cependant, c'est là qu'ils pourront en apprendre les bases.

- Maîtriser un logiciel de programmation adapté aux élèves

Environ 38% des enseignants sondés ont émis le souhait d'être formés aux outils (logiciels, robots, tablettes, etc.). En effet, il est très difficile de devoir former des élèves, sans être soi-même à l'aise avec leur manipulation, ce sentiment pouvant provoquer un rejet de cet enseignement. Ainsi, la manipulation concrète de ces outils sera nécessaire et se déroulera en deux temps : un temps de découverte¹⁸ de l'utilisation de base du logiciel Scratch ou ScratchJr (préconisé par le document d'accompagnement), mais également une phase, plutôt en fin de présentiel, mettant les enseignants en résolution de problèmes sur l'outil. Nous pourrions alors réinvestir ce que nous venons d'apprendre de la pensée algorithmique, lors d'une expérimentation, dans une situation composée de labyrinthes. Les enseignants seront alors invités à imaginer un algorithme permettant de sortir de trois labyrinthes, quelque peu différents¹⁹. Au fur et à mesure des situations, l'objectif est alors d'amener les enseignants à faire un lien entre la démarche scientifique et la pensée algorithmique. Dans le but de ne pas placer les enseignants dans une situation de frustration, leur algorithme ne permettant théoriquement pas de sortir de toute forme de labyrinthe, la présentation²⁰ et les explications²¹ de l'algorithme de Pledge (algorithme permettant de sortir de n'importe quel labyrinthe) devront être montrées.

- S'approprier une séquence d'apprentissage

Afin d'accompagner de nouveau les enseignants dans la mise en œuvre de cette initiation, un temps d'appropriation de la seconde séquence paraît pertinente. Celle-ci nécessitant cette fois des outils numériques, il conviendra de proposer, en complément, un temps de manipulation où les enseignants

¹⁸ Exemple de situation « Premiers pas », sous forme de problème : <https://scratch.mit.edu/projects/142493438/>

¹⁹ Exemple de situation élaborée, nécessitant une pensée plus poussée : <https://scratch.mit.edu/projects/142500296/>

²⁰ Exemple de situation : <https://scratch.mit.edu/projects/142501856/>

²¹ Explications liées à l'algorithme de Pledge : https://interstices.info/jcms/c_46065/l-algorithme-de-pledge

seront invités à tester ce que les élèves seront amenés à faire. Le rôle des formateurs est alors de les étayer et de répondre aux éventuelles questions. Le choix des séances à réaliser sur le support (ordinateur ou tablette) et la durée varieront alors en fonction des besoins des enseignants. Ce temps s'inscrira entre le temps de découverte des logiciels et la situation d'expérimentation, leur permettant de mettre en œuvre les premières compétences dans des situations simples. Il leur sera également proposé un accompagnement en classe, s'ils en ressentent le besoin et en font la demande.

2.2.4. Démarche

Il est reconnu que l'usage continu et unique d'un discours descendant n'est pas la démarche la plus appropriée pour l'acquisition de connaissances chez les formés, notamment lors d'une formation en petit groupe. L'enrôlement des enseignants, prenant appui sur des questionnements et l'activation de leur réflexion personnelle, est bien plus pertinent pour qu'ils construisent leurs apprentissages. On ne s'appuiera donc pas sur une méthode affirmative, mais plutôt interrogative pour les parties liées à la théorie (lexique, démarche scientifique, pensée algorithmique). On pourra alors tendre vers une technique d'animation se basant sur du brainstorming et sur une discussion avec support. En ce qui concerne les phases où l'on souhaite mettre les enseignants dans une situation de résolution de problèmes, il conviendra d'utiliser de méthode expérimentale, afin qu'ils puissent tâtonner, avancer à l'aide des essais-erreurs, et apprécier par eux-mêmes la démarche scientifique et la pensée algorithmique. Cela se déroulera de manière individuelle ou en sous-groupes, avec comme entrées, les situations de résolution de problèmes citées dans la partie précédente. Enfin, lors de l'étude des séquences proposées par 1, 2, 3 codez, c'est cette dernière organisation qui sera retenue, des sous-groupes homogènes en fonction du cycle d'enseignement étant favorable aux échanges et aux questionnements similaires. Il conviendra ici de mettre en place une méthode active, dans le but qu'ils s'approprient, par leur engagement personnel et à travers la coopération, les connaissances et les objectifs liés à leur cycle. En somme, il est important d'élaborer une démarche organisée autour de différentes méthodes, en fonction des phases de formation. Les techniques d'animation dépendent de ces méthodes, il conviendra donc d'alterner entre des organisations collectives et des groupes de travail.

Une communication verbale est à privilégier, prenant appui sur des échanges formateur / formés ou entre pairs, notamment lors des activités de groupes. Il s'agira là de créer un espace de parole et de s'assurer que tous les formés puissent s'exprimer. Essentiellement lors des phases collectives, la place du formateur aura un rôle important à jouer, tout comme l'organisation spatiale des tables, le tout pouvant être organisé en « U ». Lors des ateliers d'analyse de séquence et de manipulation en sous-groupes, il est intéressant de répartir les enseignants en îlots, le formateur circulant alors en fonction des besoins, mais restant tout de même dans l'espace visuel des formés. Il doit faire preuve de sa qualité d'écoute et d'expression, de son empathie et de son altruisme afin que cette communication soit la plus optimale possible.

La participation à cette formation étant volontaire, il est évident que l'état d'esprit des formés est optimal. Comme le montrent les résultats du questionnaire préalable, plusieurs d'entre eux avaient déjà mis en place quelques activités en classe. Cependant, tous avaient commencé par des activités dites « branchées », en se focalisant sur les compétences liées aux outils numériques, sans percevoir les objectifs transversaux. Ainsi, afin de garder les enseignants motivés et de les amener à mettre leurs pratiques en lien plus étroit avec les instructions, il faut faire preuve d'indulgence, en les accompagnant vers ces nouvelles pratiques. Il ne faut pas oublier que l'on ne peut pas former sans les stagiaires, mais qu'il ne faut pas non plus être dans la modélisation. Ainsi, il sera nécessaire de baliser le chemin à parcourir, en définissant notamment les axes de progrès, et en les explicitant.

2.3. Anticipation

L'objectif de la première partie du module de formation retenu est donc axé sur les activités débranchées et le développement de l'utilisation de démarches scientifiques. Celle-ci s'appuyant sur les séquences 1, 2, 3 codez, il est nécessaire de préparer en amont tout le matériel, afin que les formés puissent les prendre en main de manière optimale (quadrillage sur feuille A1, cartes de déplacement, etc.). Pour le second présentiel, il est également nécessaire d'anticiper au niveau du matériel, celui-ci nécessitant cette fois des outils numériques. Ainsi, en raison des manipulations souhaitées (tablettes et ScratchJr pour les cycles 1 et 2, ordinateurs et Scratch pour le cycle 3), il a été demandé aux formés de se munir, si cela est possible, du matériel adéquat (tablette avec ScratchJr d'installé et/ou ordinateur portable). Afin de prendre en compte les contraintes de chacun, trois tablettes et trois ordinateurs portables de l'inspection seront également mises à disposition. Enfin, notamment en raison des manipulations et des activités prévues sur le logiciel Scratch, il est obligatoire que, sur le lieu retenu, il y ait la possibilité de se connecter au WiFi.

Afin de mener au mieux une formation, il convient de définir un lieu qui soit le plus adapté possible. Pour que celle-ci soit la plus pragmatique possible, il est judicieux qu'elle soit dispensée dans une classe, ce qui permet ainsi d'échanger sur les organisations spatio-temporelle et matérielle des activités qui peuvent être proposées aux élèves. Ce module nécessite également la présence d'un vidéoprojecteur, afin de pouvoir projeter les supports de présentation qui ont été élaborés. Enfin, ce dernier se déroulant sur deux fins de journée, il est préférable qu'il soit dispensé au plus près du lieu de travail des formés. En effet, en fonction des horaires de chacun et de la possible présence d'APC après la classe, il convient de trouver un lieu central, afin que les enseignants puissent arriver à l'heure, sans se sentir « pressés ». Ainsi, pour le module réalisé, il a été choisi une école centrale, équipée d'un T.B.I. et de WiFi.

2.4. L'accompagnement des enseignants

Les séquences 1, 2, 3 codez ont été apportées et travaillées lors du module de formation afin d'accompagner au mieux les enseignants dans cette initiation. Contrairement aux premières suppositions, aucun enseignant du groupe n'a émis le souhait d'être accompagné en classe pour une aide quelconque. Cela est principalement dû au fait que tous étaient volontaires, à l'aise avec les connaissances, les outils, et

avaient déjà une réflexion plus ou moins avancée sur cette initiation avant le module. Pour autant, il est important de ne pas se cantonner à une seule séquence ou un seul projet. Ainsi, j'ai proposé diverses ressources qui pourraient leur être utiles, que ce soit avant ou pendant la séquence, mais également pour élaborer d'autres projets par la suite. Parmi celles-ci, on peut retrouver :

- Le site « <http://123codez.org> », qui reprend les séquences proposées par « la main à la pâte » ;
- Le MOOC « Class'Code », qui est une formation en ligne dont la première partie est liée à la programmation créative, que j'ai suivie ;
- Le site « code.org » qui propose diverses activités en ligne ;
- Le manuel CS Unplugged, qui propose des activités pour travailler la pensée algorithmique et qui a été traduit en français²².

Les modules élaborés par 1, 2, 3 Codez, proposent une séquence pour chaque cycle mettant en place l'utilisation de robots. Bien que le coût d'exploitation soit des plus raisonnables, il est indéniable que les enseignants souhaitent pouvoir avoir des informations sur les différents modèles et se former sur leur utilisation. Afin de répondre à ces demandes croissantes, il a été possible de s'en procurer auprès de la DSDEN du Doubs. Cependant, nous n'avons pu être informés que quelques jours avant le second présentiel. Celui-ci étant déjà très dense, nous avons proposé aux enseignants le souhaitant, de présenter succinctement les différents types de robots à disposition et de les manipuler après le temps de formation, afin qu'ils investissent dans du matériel adapté le cas échéant. Bien que le prêt de ce matériel aux écoles puisse être théoriquement possible dans quelques mois, ce n'est actuellement pas le cas. Nous avons donc précisé aux enseignants les démarches à effectuer auprès de CANOPE, pour emprunter des robots et des tablettes.

3. Partie 3 : Analyse

3.1. L'évaluation du module de formation

Même si l'évaluation « à chaud » d'une formation est intéressante, elle ne suffit pas à avoir un retour objectif sur l'efficacité de celle-ci, l'objectivité des stagiaires étant altérée notamment par les échanges avec le formateur. Ainsi, il est nécessaire de pouvoir la mesurer sur d'autres points plus révélateurs, à savoir : l'impact sur les pratiques des enseignants, et pour aller plus loin, sur les apprentissages des élèves.

3.1.1. A chaud

La première partie comportait des activités bien différentes des conceptions initiales, les formés s'attendant à un contenu lié beaucoup plus aux outils numériques. Bien que le fait de garder les formés enrôlés paraissait difficile pour cette raison, les liens qui ont pu être faits avec la démarche scientifique les ont intéressés au plus haut point, notamment en raison du fait que cela permettait de faire des liens clairs

²² <https://interstices.info/upload/docs/application/pdf/2014-06/csunplugged2014-fr.pdf>

avec les autres domaines. En allant plus loin vers la pensée algorithmique et en les mettant en situation de résolution de problèmes, l'intérêt suscité a également été très révélateur. Cela leur a permis de mettre des mots sur des compétences qu'ils utilisent dans la vie quotidienne et qu'ils enseignaient déjà aux élèves.

Nous pouvons également noter que tous les formés présents lors de la première partie, l'étaient lors de cette seconde, et que d'autres participants ont rejoint le module. Pour ces derniers, le support de présentation leur a été transmis, en complément de quelques informations, afin de rejoindre le groupe en ayant toutes les informations nécessaires. Les derniers échanges ont soulevé le fait que tous souhaitent approfondir le sujet et expliciter les points d'appuis et les difficultés de la deuxième séquence lors d'un troisième présentiel. Ce dernier ne pouvait être mis en place, ce module ayant été validé pour deux actions de deux heures. Cependant, cela souligne et renforce les hypothèses concernant le temps de formation nécessaire, à savoir qu'en raison du contenu dense, bien que le module fut intéressant, quatre heures semblent encore insuffisantes. Nous pouvons aller plus loin en remarquant que l'aspect lié à la créativité n'a pas été étudié en profondeur.

3.1.2. Impact sur les pratiques

A l'instar des TICE, de nombreux enseignants indiquent ne pas avoir assez de temps pour « faire de l'informatique », ces derniers pensant généralement que cet apprentissage est principalement lié aux compétences d'utilisation de l'outil. Pour autant, l'engagement des enseignants dans cette initiation fut important. Parmi les huit enseignants faisant partie du groupe de travail, trois enseignants avaient proposé des activités avant le module. Suite à la première formation, six d'entre eux ont mis en place la première séquence, les deux derniers enseignants ayant indiqué ne pas pouvoir le proposer actuellement en classe, étant engagés dans de nombreux projets. Cependant, ils ont indiqué qu'ils souhaitent tout de même continuer ce module de formation. Concernant la seconde séquence, qui est actuellement mise en œuvre dans plusieurs classes, les enseignants ont indiqué qu'ils allaient étudier plus en détails les étapes de la démarche scientifique avec les élèves. En effet, les observations en classe que je détaillerai par la suite, ont mis en avant quelques manques lors des séances et notamment le fait que les phases d'hypothèses et d'analyse n'étaient que peu abordées.

Il convient ensuite d'observer si les objectifs posés par l'enseignant sont indiqués sous forme de compétences informatiques ou si des compétences transversales, et notamment liées à la résolution de problèmes, sont mises en avant. Bien que leur progressivité soit intéressante, les séquences proposées par « 1, 2, 3 codez » privilégient l'explicitation de l'apprentissage sur la base de compétences liées à l'informatique, et ne font que peu de liens explicites vers les compétences transversales, notamment le raisonnement et la formation du futur citoyen. En effet, les objectifs des séances y sont généralement formulés comme des compétences numériques à atteindre. Ainsi, il a été intéressant de remarquer que, même si cela se voit encore chez certains formés, les compétences transversales sont mises en avant. Il est possible de prendre comme exemple : « Apprendre à planifier son action avant de la réaliser », « Donner

son avis en argumentant au sein d'un groupe » ou encore « Être capable de résoudre un problème complexe de deux étapes ».

Le troisième point intéressant est d'évaluer la capacité des enseignants à adapter les séances proposées en fonction du contexte (élèves, matériel, etc.). L'évaluation porte ainsi sur les adaptations pédagogiques réalisées d'un point de vue pragmatique. Les séquences intègrent généralement l'utilisation d'un robot dans leur seconde partie. Or, la plupart des écoles n'en sont pas équipées. Les enseignants de cycle 1 ont alors adapté les situations proposées au cycle 2 afin de proposer tout de même quelques activités branchées. Par ailleurs, deux des enseignants de cycle 2 ont émis le souhait d'avoir un accompagnement de l'ATICE quant à la possibilité d'utiliser ScratchJr sur des ordinateurs et l'ajustement lié à son utilisation.

Enfin, ces séquences sont proposées pour une année de cycle. Il est donc judicieux de s'intéresser à ce que les enseignants envisagent de proposer les années suivantes. Lors de la mise en œuvre de la seconde partie du module, un formé a d'ailleurs soulevé ce questionnement. Ayant enrôlé toute l'équipe d'école dans ce projet, deux autres enseignantes se sont déjà projetées, en se questionnant sur ce qui pourrait être mis en place lors des prochaines années, afin de rester dans une cohérence et une continuité des apprentissages. Ces deux équipes sont en train de concevoir, à partir des séquences « 1, 2, 3 codez », mais également à partir des aides et outils proposés lors du premier présentiel, de nouveaux projets transdisciplinaires (notamment en lien avec la production d'écrits).

3.1.3. Impact sur les apprentissages des élèves

La première évaluation possible au niveau de l'impact sur les apprentissages des élèves, peut être appréciée à partir de la situation de référence présentée lors du premier présentiel (annexe 5). Il est alors possible, en fin de séquence, de proposer de nouveau cette situation, afin d'analyser l'évolution des réponses proposées. Les élèves proposent-ils la même réponse, ou vont-ils privilégier une réponse plus algorithmique ? Vont-ils utiliser un code ? Les boucles seront-elles envisagées ? Pour aller encore plus loin, il serait intéressant de les mettre dans une situation similaire, voire dans un contexte sensiblement différent et plus complexe. La seconde séquence étant encore actuellement dispensée dans les classes, cette évaluation ne peut pas être réalisée. Cependant, les résultats seront intéressants, puisqu'il a été remarqué que la plupart des élèves de cycle 3 proposaient une réponse sous forme de production d'écrits. A noter que les élèves de cycle 1 ont généralement choisi d'opter pour du code à l'aide de flèches et non une dictée à l'adulte.

La seconde concerne l'usage des démarches scientifiques et de la pensée algorithmique en lien avec la résolution de problème, notamment complexe. Cependant, cette évaluation est difficile, compte-tenu du fait que les procédures de résolution sont nombreuses. Cette évaluation peut se faire « au fil de l'eau », en observant les méthodes utilisées par les élèves et en vérifiant que ces derniers ont recours à une démarche scientifique efficiente, dès lors que celle-ci est pertinente. Il est à préciser que cette évaluation peut se dérouler dans tous les domaines et à tout moment, et non uniquement dans les domaines scientifiques.

Enfin, il peut être pertinent d'apprécier l'évolution de certaines compétences transversales, en lien avec le domaine 3 sur socle commun « la formation de la personne et du citoyen ». En effet, la démarche de projet et le travail collaboratif qui sont mis en avant, favorisent notamment lors des débats au sein d'un groupe, le développement de nombreuses de ces compétences. Ainsi, il sera judicieux d'apprécier, au fur et à mesure du temps, le respect des autres, des règles, des droits, le sens de l'engagement, l'initiative, mais également tout ce qui est lié à la persévérance de l'effort face aux erreurs ou à l'échec. Cette initiation à la programmation pourra permettre d'amener les élèves à résoudre des conflits grâce au langage et à l'argumentation, ce qui, in fine, pourra favoriser le climat lors de travaux de groupes et ainsi permettre aux élèves de construire davantage leurs savoirs en collaboration.

Ces trois aspects de l'évaluation de la formation au niveau de l'impact des apprentissages des élèves sont à réaliser quotidiennement, en appréciant l'évolution des pratiques et de l'attitude des élèves en classe. L'enseignant sera alors le plus à-même à évaluer ce point.

3.2. Le suivi en classe

Bien qu'aucun enseignant n'aie émis le souhait d'avoir un accompagnement, je me suis permis de demander à plusieurs d'entre eux de pouvoir assister à quelques séances, notamment en cycles 2 et 3. L'objectif était de rester attentif à l'évolution des pratiques dans ces classes et à ce qui est proposé aux élèves pour faire évoluer la proposition de formation. Il paraît également opportun d'aider ces précurseurs à accéder à un niveau plus élevé de pratique pédagogique au niveau de la démarche scientifique et de la pensée algorithmique. Dans le cas d'enseignants novices sur le sujet et/ou peu à l'aise avec les outils numériques, un accompagnement réalisé par l'ATICE ou par un conseiller pédagogique de la circonscription serait à mettre en place, que ce soit pour les premières activités débranchées ou pour les activités utilisant les outils numériques.

3.2.1. Module – Partie 1

La première séance observée s'est déroulée dans une classe de CE2-CM1 située dans l'école d'un petit village faisant partie d'un RPI. Les élèves ayant déjà travaillé sur certaines compétences liées à l'initiation à la programmation, le choix de la séquence s'est porté sur celui de cycle 3. L'observation correspondait à la 3^{ème} séance de la première séquence, dont l'objectif final était d'être capable d'envoyer un message codé à un véhicule (module lunaire) afin de le piloter à distance. La séquence amenait les élèves au fur et à mesure à encoder et à décoder des messages dans un langage binaire (« 0 » ou « 1 » étant les deux seuls caractères autorisés), langage utilisé par les ordinateurs. Lors de la séance précédente, ils ont été amenés à trouver comment encoder un message à l'aide de tous les nombres (01 correspondait à la lettre A, 02 à la lettre B, etc.). Une nouvelle contrainte (encodage en binaire) apparaît alors durant la séance observée. Les élèves étaient amenés à user de démarche scientifique, afin de trouver la liste des combinaisons d'encodage (4 possibilités en 2 bits, à savoir 00, 01, 10 et 11, annexe 6). L'observation a permis de remarquer que l'enseignante n'avait pas explicité en détail les étapes de cette démarche. Ainsi,

les élèves n'émettaient pas d'hypothèses et listaient, avec un choix de raisonnement et d'organisation trop léger, les nombres possibles. Pour l'encodage en 2 bits, cela a été relativement facile, mais la procédure fut bien plus compliquée pour l'encodage en 3, 4, puis 5 bits (annexe 7), où une organisation plus poussée, nécessitant une réflexion et une bonne anticipation, était nécessaire. Ainsi, l'observation des méthodes de chacun des groupes (dès l'encodage en 3 bits), a permis de faire émerger des techniques permettant de lister, sans en oublier, les nombres possibles. La procédure retenue par la classe, après des échanges oraux, a été de les lister dans l'ordre croissant. Par la suite, une reconnaissance de schéma, liée à la pensée algorithmique a émergé : les élèves se sont rendus compte que le nombre de possibilités doublait à chaque fois (4 organisations pour 2bits, 8 choix pour 3bits, 16 pour 4bits, etc.). Ainsi, ils ont pu, sans avoir à les lister de nouveau, donner le nombre de combinaisons pour 6, 7 et 8bits, et choisir l'encodage le plus adapté à ce que l'on souhaitait envoyer comme information. En effet, pour encoder les jours de la semaine, un encodage en 3bits suffit (8 possibilités), alors que pour les mois, un encodage en 4bits est nécessaire. Enfin, ils ont proposé d'utiliser un encodage en 5 bits, afin d'avoir un nombre suffisant pour encoder les 26 lettres de l'alphabet et garder 6 autres combinaisons pour des caractères spéciaux (comme l'espace entre deux caractères notamment). Cette séance fut très riche, et bien que les étapes de la démarche scientifique n'aient pas été explicitées au préalable, il a été nettement observé que les élèves avaient déjà développé des techniques intéressantes de raisonnement.

Je suis également allé observer une séance en cycle 2, dans une classe de CP-CE1 située dans une petite ville. Celle-ci correspondait à la 4^{ème} séance de la séquence dont l'objectif était d'amener les élèves à découvrir et comprendre le système de pixellisation d'une image en noir et blanc, en utilisant de nouveau le langage binaire, mais de manière plus implicite. Dans cette séquence, les élèves se mettent à la place d'un héros qui s'éveille sur une île, dans un monde inconnu et qui doit résoudre des énigmes, encoder et décoder des messages ou des images et programmer les déplacements d'un sous-marin, ce qui lui permettra de retourner chez lui. Il a été remarqué que les élèves sont très investis par le projet. Durant la séance observée, le héros devait encoder un message sur la plage, à l'aide de cailloux noirs ou blancs, permettant aux oiseaux magiques, qui ne parlent pas le français, de comprendre qui ils devaient aller chercher, en l'occurrence un magicien. Les élèves proposent rapidement de réaliser sur la plage un dessin à l'aide des cailloux. L'enseignant les amène à choisir le « chapeau » du magicien, proposé par la séquence. Puis ils ont été invités à encoder cet objet, à l'aide d'un quadrillage et du modèle du chapeau. L'idée était de colorier certaines cases en noir, représentant les cailloux noirs disponibles sur la plage. Les essais-erreurs observés lors de cette séance ont été intéressants. Les élèves ont été amenés à reproduire en « calquant », et en faisant le choix de colorier certaines cases ou non. Pour certains élèves, l'abstraction fut problématique, car ils ne comprenaient pas pourquoi ils devaient colorier toute la case (annexe 8). Une fois que leur production répondait aux critères de réalisation, elle fut analysée. Certains d'entre eux ont souhaité adapter leur choix, afin que le dessin encodé ressemble davantage au modèle.

3.2.2. Module – Partie 2

La seconde séquence est actuellement mise en œuvre dans les classes. J'ai pu tout de même assister à une séance, dans la classe de CE2-CM1-CM2 présentée dans la partie précédente. Il s'agit d'une séance de familiarisation au logiciel « Scratch ». Il est pertinent de se questionner sur la démarche retenue pour l'appropriation de cet outil. Les élèves, après avoir manipulé librement durant quelques minutes, sont placés dans sept situations de résolution de problèmes (annexe 9). Cette classe faisant partie de celles ayant déjà manipulé le logiciel avant la formation, ces situations sont rapidement proposées aux élèves. Celles-ci ont été pensées dans un ordre logique d'apprentissage, sous forme de progressivité. L'objectif principal est d'amener les élèves à user de pensée algorithmique dans un contexte inhabituel. Des échanges opportuns ont eu lieu sur le fait que le logiciel faisait exactement ce qu'on lui demandait de faire. Ainsi, l'enseignante a insisté sur le rôle important de l'utilisation du langage oral lors de la construction de l'algorithme, en indiquant qu'il fallait « mettre des mots » sur ce que l'on souhaite que le personnage (appelé « lutin » sur Scratch) fasse. Bien que les élèves aient déjà été familiarisés avec le logiciel, l'observation a permis de remarquer que la navigation des élèves sur celui-ci était aisée. Par ailleurs, les possibilités du logiciel sont conséquentes. Il a été pertinemment choisi de se focaliser sur les quatre principaux menus, à savoir « mouvement », « évènements », « apparence », et « contrôles ». En complément, une explicitation de quelques-unes des possibilités pour chaque partie a été effectuée par les élèves. Suite à l'accompagnement dispensé lors de la première séance, une évolution a été observée concernant la pratique des démarches scientifiques. En effet, l'enseignante a engagé les élèves à émettre des hypothèses et à expliciter le résultat qu'ils pensaient atteindre. Ces différents échanges ont permis de résoudre les bugs rencontrés, notamment lors de l'exercice 7 (annexe 10), en insistant sur le moment auquel on souhaitait remettre le « lutin » au centre de la scène. La position utilisée (à savoir $x : 0$ et $y : 0$) est par ailleurs judicieusement amenée par la séquence. Le travail en amont sur les pixels (Cf : séquence 1), a permis aux élèves de mettre du sens et de comprendre à quoi correspondaient les valeurs données pour « x » et « y » et le lien avec le nombre de pas que fait le lutin. Enfin, la progressivité des exercices amène les élèves à découvrir les « boucles » et à en comprendre les intérêts. Les élèves ont pu se rendre compte, après avoir reconnu un schéma, que l'utilisation de boucles permettait d'avoir un code très simple à lire, mais qu'il fallait alors une organisation bien réfléchie.

3.3. Constats et adaptations possibles

Je vais désormais m'attacher à évoquer quelques constats faits lors des formations et proposer quelques adaptations possibles à ce module.

3.3.1. Conception, élaboration, anticipation

L'anticipation réalisée a permis de dispenser le module de formation dans une école qui pouvait proposer du WiFi. Nous avons pour autant été confrontés à un problème qui aurait pu être évité, à savoir l'accès à cette connexion. En effet, les enseignants ont été invités à rejoindre le site de l'outil Scratch en

ligne durant le temps de formation, en se connectant au WiFi de l'école. Cependant, bien que cet accès fut rapide, nous avons perdu de précieuses minutes, notamment en raison du fait que la clé WiFi comportait de nombreux caractères. Afin de ne pas réduire la sécurité du réseau de l'école ou de modifier son fonctionnement, le changement vers un mot de passe sensiblement plus simple ne paraît pas la réponse la plus adaptée. Il aurait été opportun de se renseigner sur l'emplacement de la box, et d'activer ce qui est communément appelé le « WiFi facile », permettant à tout ordinateur de se synchroniser à ce réseau sans aucun mot de passe à entrer durant un laps de temps assez court. Par ailleurs, et pour optimiser davantage le temps, cette synchronisation aurait dû se faire durant le temps d'accueil.

3.3.2. Lors de la mise en œuvre

Bien que cela fut un point fort du premier présentiel, l'utilisation de la démarche scientifique comme point d'appui à la résolution de problème, notamment pour les problèmes complexes, n'a pas été clairement mise en œuvre dans les classes. En effet, malgré le fait que les formés aient acquis et compris les intérêts du recours à une telle démarche, ils ont confirmé ce que j'ai pu remarquer lors des visites en observation, à savoir que cette démarche, ainsi que ses étapes, n'ont pas été explicitement travaillées. Ainsi, il serait pertinent de proposer, lors du premier présentiel, des exemples plus concrets. Nous pourrions mettre en place une expérimentation à partir d'une problématique liée aux changements d'états de la matière et mettre des mots sur chaque étape, en donnant un support écrit les récapitulant. D'autres situations, bien plus ludiques, pourraient être proposées si le temps était plus conséquent. Nous pourrions par exemple imaginer une situation dans laquelle les formés tenteraient de répondre à cette problématique : « Comment faire tomber un œuf cuit du 3^{ème} étage sans qu'il ne soit fissuré et en n'utilisant des objets dont le poids total ne dépasse pas celui de l'œuf ? ».

L'expérimentation du second présentiel, où les enseignants étaient invités à trouver un algorithme permettant au robot de sortir de trois labyrinthes différents, est une situation qui est, selon moi, à conserver. En effet, on a pu observer clairement les enseignants mettre des mots et se rendre compte de ce qu'était la pensée algorithmique au fur et à mesure de l'élaboration de leur algorithme. Contrairement à la situation de découverte, les premiers questionnements ont été : « Peut-on voir les trois labyrinthes ? », ce qui amène à dire qu'ils ne souhaitent pas chercher comment sortir du premier labyrinthe, mais déjà anticiper la réponse et l'adapter avant même de lancer leur programme. En d'autres termes, cela permet d'anticiper les déplacements du robot en fonction de notre algorithme pour le deuxième labyrinthe. Cela a de nouveau été remarqué par la suite, lorsqu'ils étaient invités à coder le changement d'arrière-plan : au lieu de vouloir « passer à l'arrière-plan numéro deux », il a été privilégié un questionnement sur la possibilité de « passer à l'arrière-plan suivant ». Par ailleurs, ils ont clairement reconnu une situation qui leur a posé problème lors de la découverte de l'outil, à savoir « donner un point de départ au robot et assigner un arrière-plan » avant même de commencer à coder. Au niveau du code, les enseignants ont proposé instinctivement, pour la plupart, l'utilisation de boucles pour les déplacements.

3.4. Un exemple de productions : Le « Pixel Art »

Lors de l'analyse de l'impact sur les pratiques des enseignants, nous avons évoqué la production de séances ou de séquences par les enseignants. A partir des séquences proposées, il est possible de mettre en place de nombreux projets transdisciplinaires. Il a été intéressant de remarquer, dans plusieurs classes de cycle 2, que des liens ont été réalisés avec les arts, particulièrement avec le « Pixel Art » (annexe 11). Pour aller plus loin et favoriser la créativité, il serait opportun de proposer une séance où les élèves imagineraient, à l'instar de la séance sur le chapeau du magicien, une image pixellisée. Cette dernière pourrait d'ailleurs se baser sur une carte en géographie, des arts rupestres, une plante ou un animal en sciences, etc. Une séance, en cycle 3, sur l'encodage de formes pixellisées très simples (sur un quadrillage 5 x 5 par exemple) pourrait être intéressante. L'algorithme serait alors encodé en binaire. Par ailleurs, et pour aller plus loin au niveau des algorithmes, nous pourrions demander aux élèves d'élaborer une suite d'instructions sur Scratch, permettant de reproduire, en un clic, une production. Celle-ci pourrait être différenciée en fonction des capacités des élèves.

Le module de formation dispensé a été d'une manière générale positif. Quelques ajustements sont à réaliser, notamment au niveau de l'anticipation matérielle. Cependant, le contenu paraît pertinent au regard des échanges avec les enseignants, des observations et des constats réalisés. Il est à noter que la créativité, qui est l'un des piliers, n'a pas été développée dans les détails durant ce module en raison du temps alloué. Si une formation plus longue était possible, il serait intéressant de proposer du contenu en ce sens.

4. Partie 4 : « Pour aller plus loin »

4.1. Formation initiale

L'arrivée de cette initiation à la programmation pose question quant à la place qui lui est attribuée dans la formation initiale. J'ai ainsi échangé avec M. FALLOT, responsable des TICE au niveau de l'ESPE de notre académie, qui a pu m'informer sur le projet académique quadriennal. Ce dernier, ayant démarré en septembre 2013, n'intégrait pas cette initiation, compte-tenu du fait que le programme en vigueur à ce moment-là, n'y faisait aucune référence. Cependant, la planification des cours auprès des M1 et des M2 a été adaptée pour introduire un chapitre spécifique sur l'enseignement du codage au primaire. Pour aller plus loin, il serait intéressant d'intégrer ce chapitre au sein de plusieurs domaines, afin d'entrer dans une démarche de projet, comme cela est préconisé dans le programme. En effet, comme nous pouvons le voir actuellement, la grande majorité des projets est étroitement articulée avec les mathématiques et la structuration de l'espace. Or, même si les instructions officielles et la plupart des outils mettent en avant les déplacements, il serait pertinent d'élargir vers des domaines qui sont moins scientifiques. Que ce soit en classe ou à l'ESPE, pourquoi ne pas imaginer une séquence, permettant de programmer l'organisation quotidienne d'un boulanger ou d'un médecin sur le logiciel Scratch ? Cela permettrait de mettre clairement en pratique la pensée algorithmique des élèves.

4.2. Formation continue – Plan Annuel de Formation

En tant que formateur, nous avons accès à des journées de formation, dont l'une est spécifique aux usages pédagogiques du numérique. L'initiation à la programmation a été abordée durant ce temps de formation, ce qui m'a permis de prendre contact et d'échanger avec les représentantes de la DANE des 1^{er} et 2nd degrés et d'échanger à propos de la progressivité de cet apprentissage. Cependant, les enseignants n'ont pas accès à cela. Ainsi, il serait possible d'insérer un module au niveau du plan de formation des circonscriptions, tout comme l'a été la « formation au numérique », dispensée lors de l'année scolaire 2014-2015.

Une expérimentation a été proposée, lors de l'année scolaire 2015-2016, à deux CPC novices sur le sujet. L'objectif était de tester le MOOC Class'Code que j'ai également suivi durant cette année scolaire. Cette formation développe pertinemment l'un des aspects qui n'a pas été étudié en profondeur, à savoir la « programmation créative ». L'un des participants a indiqué que, d'une part, cette formation était culturellement intéressante, mais que d'autre part les réinvestissements professionnels directs lui semblaient compliqués, point de vue que je partage. Il indique qu'il aurait été souhaitable d'avoir plus de temps en présentiel (ce que les formés du module ont également souligné). Aucune suite n'a, pour l'heure et à ma connaissance, été donnée concernant cette formation à distance. Ainsi, il serait possible d'imaginer une adaptation du module au niveau du plan annuel de formation de deux ou trois journées, prenant également appui sur ce MOOC, notamment sur le volet « créativité » qui n'a pu être qu'abordé, et en faisant des liens avec les autres domaines d'apprentissage. C'est en ce sens qu'une demande a été déposée pour l'année scolaire 2017-2018.

4.3. Parcours M@gistère

L'offre de parcours sur la plateforme de formation M@gistère ne propose pas de parcours lié à la programmation pour le primaire, alors qu'il en existe plusieurs pour le secondaire. Dans ces derniers, on peut remarquer que les liaisons entre les disciplines du collège ne sont pas réalisées et que ces parcours sont à destination d'enseignants de mathématiques pour l'un, et de technologie pour l'autre. Par ailleurs, même si la partie liée aux attentes institutionnelles est intéressante, peu de liens sont faits avec les compétences transversales. En effet, la majorité du contenu porte vers l'apprentissage des outils, et notamment de Scratch. Afin de partager l'ensemble de mes recherches, j'ai demandé l'ouverture d'un parcours M@gistère intitulé « Initiation à la programmation au cycle 3 : développer la pratique des démarches scientifiques et la pensée algorithmique », qui sera soumis à validation. Actuellement en cours de conception, celui-ci a pour but de partager, de former et d'informer les ATICE du Doubs sur cette initiation à la programmation, et notamment sur l'élaboration d'une formation sur le sujet. Le parcours reprendra le module de formation que j'ai proposé, du point de vue du formateur, et sera adapté en fonction de la mise en œuvre et des retours des enseignants ayant participé au groupe. Il devrait également intégrer mes diverses recherches et les documents qui ont servi de support au module.

4.4. Partenariats

En complément des nombreux ateliers de formation ainsi que du prêt possible de leur matériel (mallette scientifique, tablettes et/ou robots, etc.), et après avoir expérimenté plusieurs projets en partenariat avec ma classe, CANOPE reste un partenaire privilégié. Suite à un atelier que j'ai suivi sur l'initiation à la programmation, la formatrice a souligné cette possibilité d'aide et d'accompagnement au sein des classes, afin de proposer des projets en classe, et notamment en lien avec la créativité. Malgré tout, comme cela a pu être précisé au niveau des parcours M@gistère, il a été dommage de voir que le contenu était essentiellement focalisé sur les outils numériques et non sur les réels apprentissages que l'on souhaite atteindre avec les élèves.

Il serait également très intéressant que des partenariats soient réalisés avec le collège, ce serait notamment un moyen concret de participer au renforcement de la liaison École-Collège, et d'ouvrir cette possibilité aux professeurs enseignant dans d'autres domaines que les mathématiques, les sciences ou la technologie. On pourrait alors imaginer la création d'une galerie interactive permettant aux élèves du primaire de découvrir les professeurs du collège ou les lieux. Inversement, ces derniers pourraient alors proposer une carte d'identité interactive les présentant. Il serait également possible et intéressant, en lien avec la découverte du patrimoine local, d'élaborer la présentation des lieux historiques propres au village ou à la ville.

Conclusion

L'initiation à la programmation n'a pas pour objectif d'amener les élèves à être capable de programmer dans un langage compliqué, mais de développer l'usage des démarches scientifiques et la pensée algorithmique, qui seront utiles dans la vie quotidienne du futur citoyen. Les situations que l'on propose aux élèves doivent donc aller en ce sens. Faisant désormais partie du programme de l'école élémentaire de manière explicite, il convient de former les enseignants du primaire à travers tous les aspects qui la composent. Bien que les enseignants novices fassent partie de ce que l'on appelle les « digital natives » et aient suivi un module à l'ESPE, cette formation se doit d'être disponible pour tous les personnels. Le module de formation proposé dans ce document est principalement axé sur le raisonnement, l'un des piliers de la programmation. Dans tous les domaines, il est possible de travailler cette compétence. Lorsque nous mettons en place des séances dans le but que les élèves développent des stratégies, que ce soit en production d'écrits, en lecture ou en analyse de documents, ne les mettons-nous pas déjà dans une logique de programmation ?

Néanmoins, le développement du raisonnement n'est pas le seul objectif. L'idée est également de s'appuyer sur un autre pilier, que le MOOC Class'Code met également en avant, à savoir la programmation créative. Cette formation la définit ainsi : « La programmation créative, c'est tout simplement apprendre à programmer de façon ludique, pour s'amuser, et pour créer. » L'objectif est de faire acquérir aux élèves les connaissances et compétences attendues en produisant, de manière ludique. Pour aller en ce sens, que ce

soit par un module en aval ou lors d'une formation plus longue, il serait possible d'approfondir le développement des compétences liées à la créativité des élèves, autour de projets transdisciplinaires en lien avec les apprentissages quotidiens.

Le développement de la pensée réflexive de l'élève peut être favorisé grâce à la programmation. Il est courant de voir que les élèves ont tendance à se fier aux préjugés et à leurs premières impressions. Il est alors très difficile pour eux de remettre en cause leurs perceptions et leurs jugements. Ainsi, cette initiation peut permettre, grâce au rôle important du langage oral et aux interactions entre pairs, de se questionner quant à la validité de leurs premières impressions, et notamment des hypothèses émises lors de la résolution de problèmes. Dans ce type de travail, le statut de l'erreur est particulier. Elle devient une ressource, un repère et il faut tout mettre en œuvre pour aller contre. La validation de leur raisonnement par l'outil est un point d'appui non négligeable car, si la démarche a correctement été réalisée, les élèves se rendront compte et seront conscients de leurs erreurs, puisque le personnage ou le robot ne fait que concrétiser leur pensée.

Enfin, il est intéressant de remarquer que cet apprentissage est en parfaite harmonie avec l'idée générale des nouveaux programmes et particulièrement avec le fait que cela demande d'enseigner plus explicitement. Cet enseignement, nécessitant une utilisation poussée du langage oral, se réalise tout au long des apprentissages, que ce soit lors du travail par groupes ou en groupe classe. L'enseignant doit expliciter aux élèves tout ce qui pourra permettre de les mettre dans des situations de clarté cognitive, au niveau du « Pourquoi ? », du « Comment ? » ou même au moment de l'institutionnalisation. Mais cet enseignement ne s'arrête pas là. Il conviendra de faire participer activement les élèves à cette explicitation, en expliquant à l'enseignant, qui les guidera avec de simples questions comme « Comment fais-tu ? », ou en échangeant avec leurs pairs.

LISTE DES ANNEXES

ANNEXE 1 : QUESTIONNAIRE

ANNEXE 2 : RESULTATS DU QUESTIONNAIRE

ANNEXE 3 : INFOGRAPHIE - LES COURS DE CODAGE EN EUROPE

ANNEXE 4 : NOUVEAU REFERENTIEL DES TICE (UNIQUEMENT LA PARTIE LIEE A LA PROGRAMMATION)

ANNEXE 5 : SITUATION DE REFERENCE

ANNEXE 6 : SUPPORT DES ELEVES – LISTING DES COMBINAISONS POSSIBLES EN 2 BITS

ANNEXE 7 : SUPPORT DES ELEVES – LISTING DES COMBINAISONS POSSIBLES EN 4 BITS

ANNEXE 8 : PRODUCTIONS D'ELEVES SUR L'ENCODAGE D'IMAGE

ANNEXE 9 : TACHES PROPOSEES AUX ELEVES (CYCLE 3 – SEQUENCE 2 – SEANCE 1)

ANNEXE 10 : BUGS PRODUITS PAR LES ELEVES (CYCLE 3 – SEQUENCE 2 – SEANCE 1)

ANNEXE 11 : PRODUCTIONS D'ELEVES EN « PIXEL ART »

L'initiation à la programmation

*Obligatoire

Votre profil

1. Année de formation initiale *

2. Dans quel(s) cycle(s) enseignez-vous ? *

Plusieurs réponses possibles.

☐ Cycle 1

☐ Cycle 2

☐ Cycle 3

3. Avez-vous déjà suivi une formation sur l'initiation à la programmation à l'école ? *

Une seule réponse possible.

☐ Oui

☐ Non

4. Si oui, pouvez-vous préciser

Plusieurs réponses possibles.

☐ Formation initiale

☐ Plan de Formation en circonscription

☐ Choix volontaire (Inscription au Plan Annuel de Formation)

☐ Choix volontaire (hors Education Nationale)

5. Avez-vous des références théoriques sur le sujet ? *

Si oui, précisez

6. Pour vous, sa mise en œuvre est : *

Une seule réponse possible.

1 2 3 4 5

Très difficile ☐ ☐ ☐ ☐ ☐ Aisée

Connaissances et compétences

7. Pensez-vous pouvoir différencier "Code", "Algorithme" et "Programmation" ? *

Une seule réponse possible.

☐ Oui, je peux les définir clairement

☐ Oui, je pense pouvoir les différencier

☐ Non, même si ces termes ne me sont pas inconnus

☐ Non, au moins l'un d'entre eux m'est totalement indéfinissable

8. A votre avis, à quoi correspondrait la "programmation" dans les nouveaux programmes ? *

Une seule réponse possible par ligne.

	Tout à fait d'accord	Plutôt d'accord	Plutôt pas d'accord	Pas du tout d'accord
Apprendre à lire un langage de programmation.	☐	☐	☐	☐
Écrire du code (avec des mots).	☐	☐	☐	☐
Trouver les étapes permettant de résoudre automatiquement un type de problème.	☐	☐	☐	☐
Programmer un robot.	☐	☐	☐	☐
Faire un programme informatique.	☐	☐	☐	☐
Décomposer une tâche complexe en tâches simples.	☐	☐	☐	☐
Traiter des informations automatiquement.	☐	☐	☐	☐

9. Pour vous, quels sont les bénéfices attendus de cette initiation ? *

Une seule réponse possible par ligne.

	Tout à fait d'accord	Plutôt d'accord	Plutôt pas d'accord	Pas du tout d'accord
Mieux maîtriser l'outil informatique	☐	☐	☐	☐
Former les élèves à d'autres compétences du socle au moyen d'un outil ludique	☐	☐	☐	☐
Permettre la découverte des concepts fondamentaux de l'informatique (langage, information, algorithmique, machine)	☐	☐	☐	☐
Apprendre à travailler en équipe	☐	☐	☐	☐
Acquérir une culture numérique	☐	☐	☐	☐
Former des futurs programmeurs potentiels	☐	☐	☐	☐
Développer les compétences en résolution de problèmes	☐	☐	☐	☐
Former les futurs citoyens	☐	☐	☐	☐
Démystifier l'informatique	☐	☐	☐	☐
Développer la créativité des élèves	☐	☐	☐	☐

10. De quoi pensez-vous avoir forcément besoin pour commencer cet apprentissage ? *

Une seule réponse possible.

- ☐ Ordinateurs (fixes ou portables)
- ☐ Tablettes
- ☐ L'un ou l'autre
- ☐ Aucun des deux, je peux commencer sans ces outils

Action de formation

Souhaiteriez-vous qu'une formation vous soit proposée :

11. Au Plan Annuel de formation (inscription volontaire) *

Une seule réponse possible.

- ☐ Oui, je m'inscrirais volontiers !
- ☐ Pourquoi pas, à voir ...
- ☐ Non, je sais déjà tout ce qu'il y a à savoir !
- ☐ Non, cela ne m'intéresserait pas.

12. Au niveau du plan annuel de la circonscription *

Une seule réponse possible.

- ☐ Oui, ce serait intéressant !
- ☐ Pourquoi pas, à voir ...
- ☐ Non, je sais déjà tout ce qu'il y a à savoir !
- ☐ Non, cela ne m'intéresserait pas.

13. Quelles seraient vos attentes si une formation était proposée ? *

14. Avez-vous déjà mis en place des séances de "programmation" ? *

Une seule réponse possible.

- ☐ Oui Passez à la question 15.
- ☐ Non Passez à la question 20.

Votre séance ou votre séquence

15. Quelles ont été les principales difficultés que vous avez rencontrées ? *

16. Le langage oral a-t-il eu une place importante ? * (en collectif, entre pairs, etc.)

Une seule réponse possible.

- ☐ Oui
- ☐ Non

17. Si oui, à quel(s) moment(s) ?

--	--

18. Quelles ont été les principales compétences travaillées ?

19. Quel est votre retour sur cette séquence ? *

Une seule réponse possible.

- ☐ Positif, je pense réitérer
- ☐ Plutôt négatif

Arrêtez de remplir ce formulaire.

Vous n'avez encore rien engagé sur le sujet

20. Envisagez-vous, à l'heure actuelle, de mettre en place des activités liées à la programmation ? *

Une seule réponse possible.

- ☐ Oui
- ☐ Non

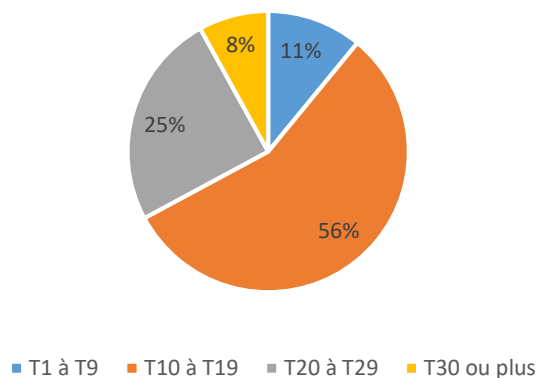
21. Si une formation sur le sujet vous était proposée, l'envisageriez-vous davantage ? *

Une seule réponse possible.

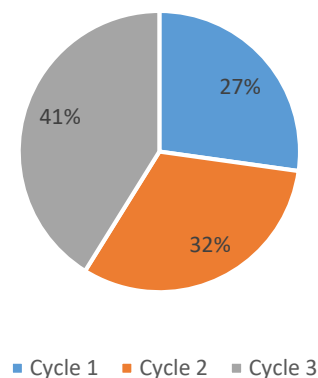
- ☐ Oui
- ☐ Non

L'initiation à la programmation

1. Ancienneté

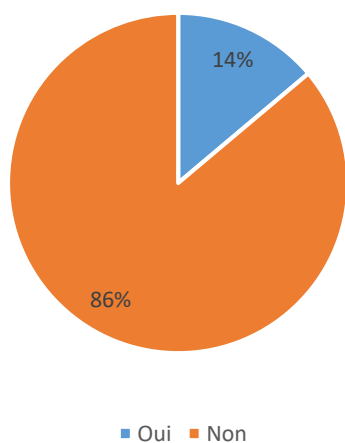


2. Dans quel(s) cycle(s) enseignez-vous ?

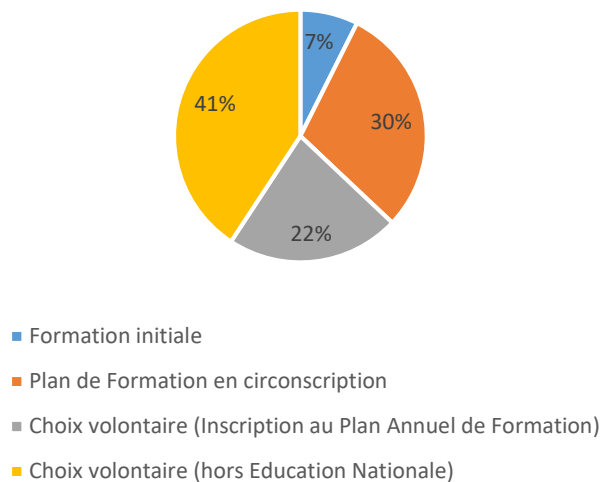


Votre profil

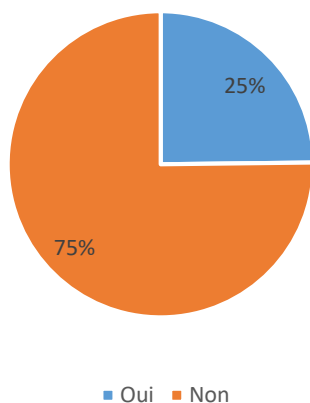
3. Avez-vous déjà suivi une formation sur l'initiation à la programmation à l'école ?



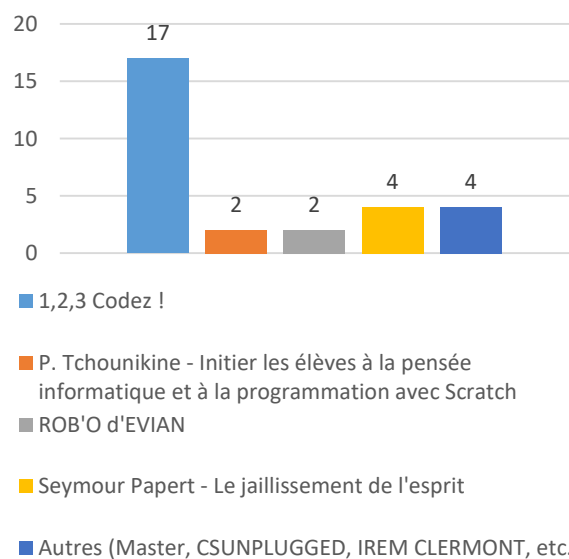
4. Si oui, pouvez-vous préciser :



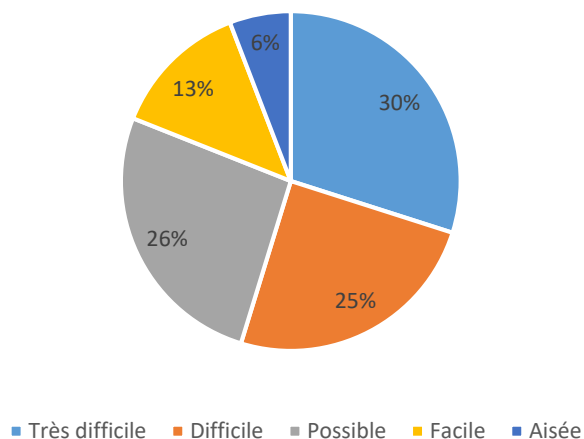
5. Avez-vous des références théoriques sur le sujet ?



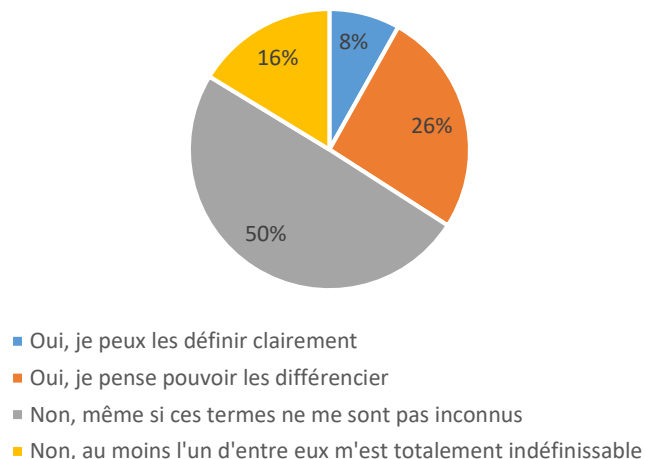
5b. Si oui, lesquelles ?



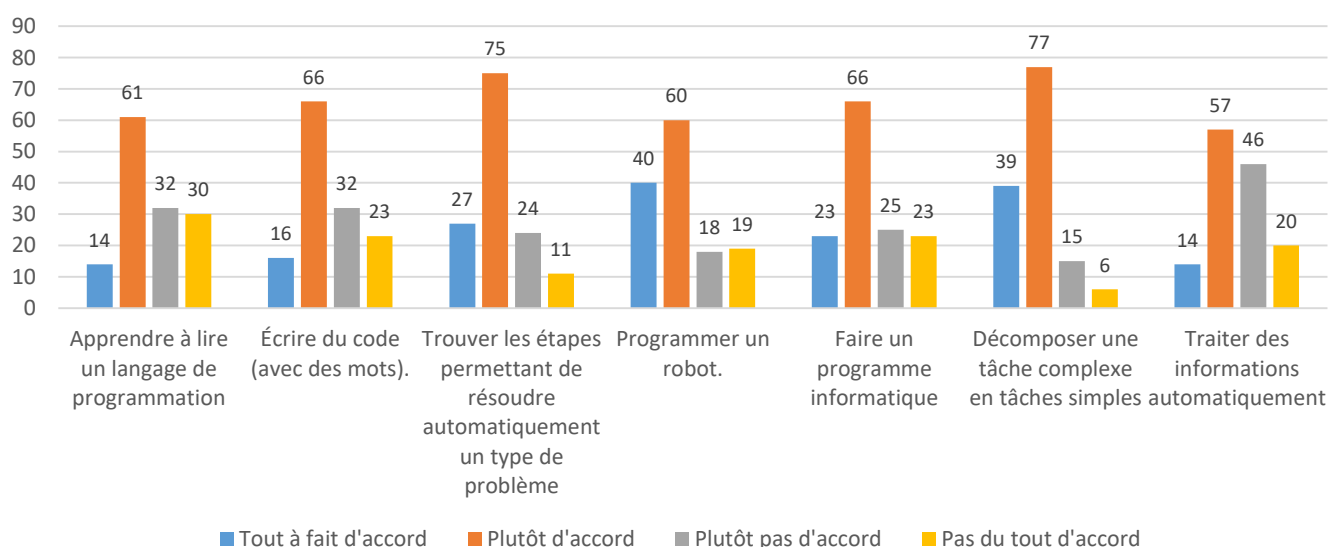
6. Pour vous, sa mise en œuvre est



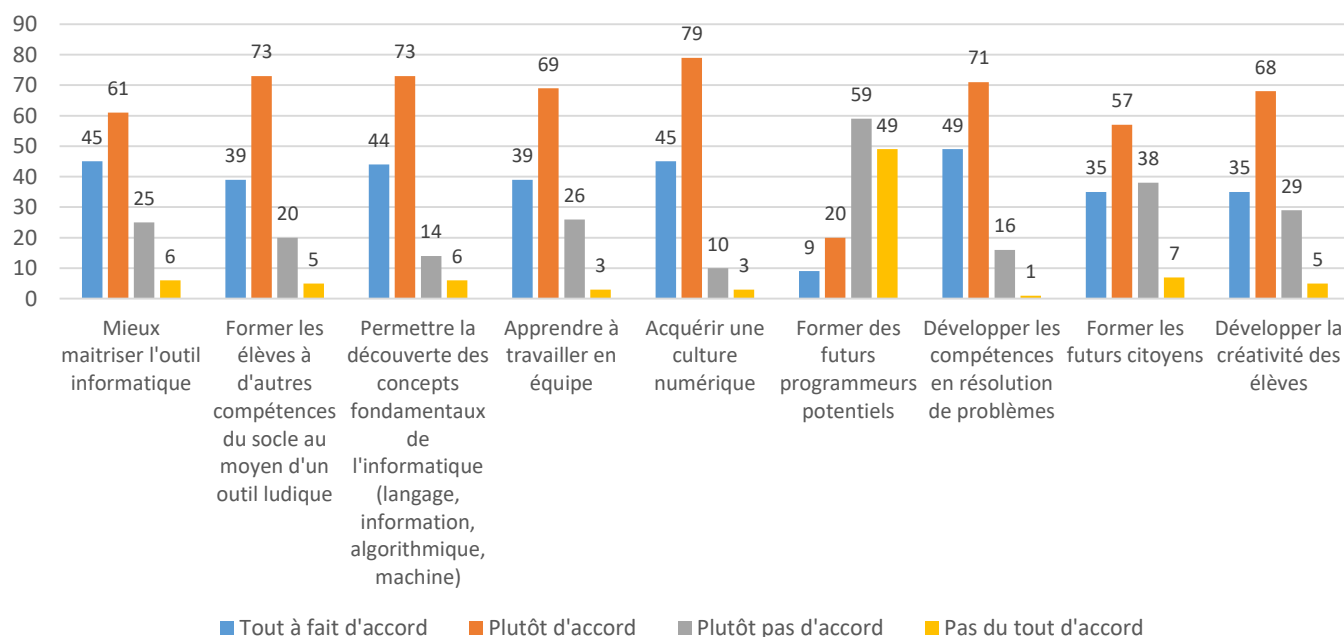
7. Pensez-vous pouvoir différencier "Code", "Algorithme" et "Programmation" ?



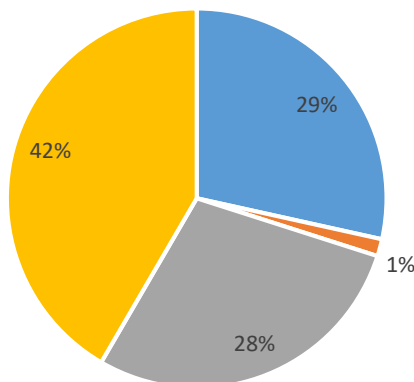
8. A votre avis, à quoi correspondrait la "programmation" dans les nouveaux programmes ?



9. Pour vous, quels sont les bénéfices attendus de cette initiation ?

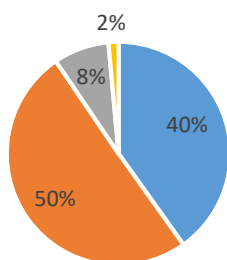


10. De quoi pensez-vous avoir forcément besoin pour commencer cet apprentissage ?



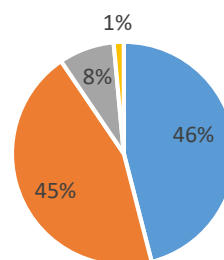
■ Ordinateurs (fixes ou portables) ■ Tablettes ■ L'un ou l'autre ■ Aucun des deux, je peux commencer sans ces outils

11. Souhaiteriez-vous qu'une formation vous soit proposée au plan annuel de formation ?



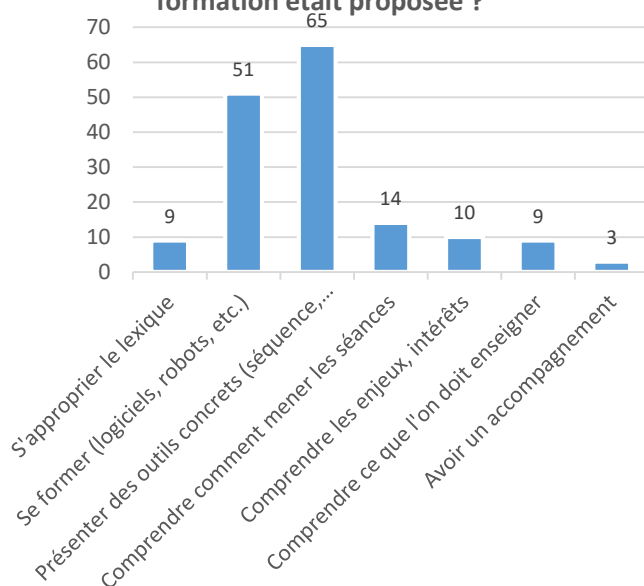
■ Oui, je m'inscrirais volontiers !
 ■ Pourquoi pas, à voir ...
 ■ Non, cela ne m'intéresserait pas.
 ■ Non, je sais déjà tout ce qu'il y a à savoir !

12. Souhaiteriez-vous qu'une formation vous soit proposée au plan annuel de la circonscription ?

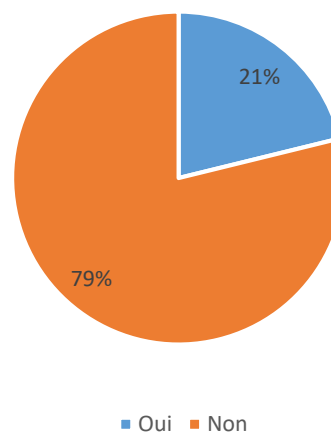


■ Oui, ce serait intéressant !
 ■ Pourquoi pas, à voir ...
 ■ Non, cela ne m'intéresserait pas.
 ■ Non, je sais déjà tout ce qu'il y a à savoir !

13. Quelles seraient vos attentes si une formation était proposée ?



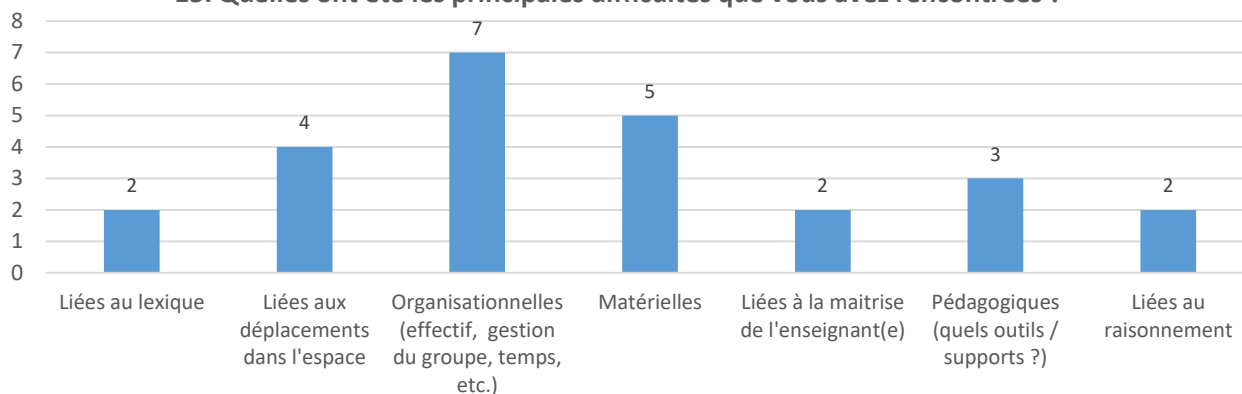
14. Avez-vous déjà mis en place des séances de "programmation" ?



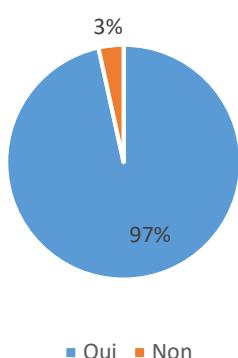
■ Oui ■ Non

Vous avez mis en place des séances sur la "programmation"

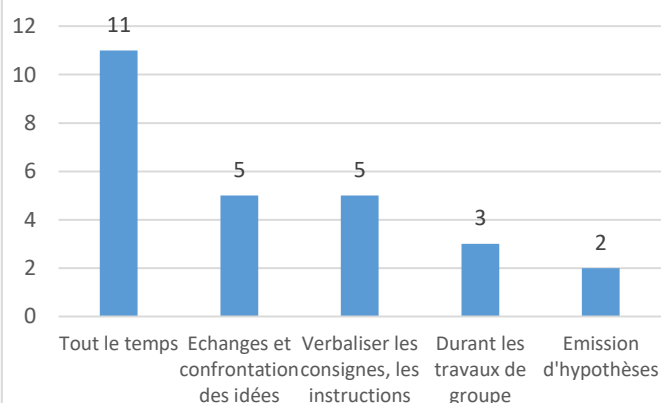
15. Quelles ont été les principales difficultés que vous avez rencontrées ?



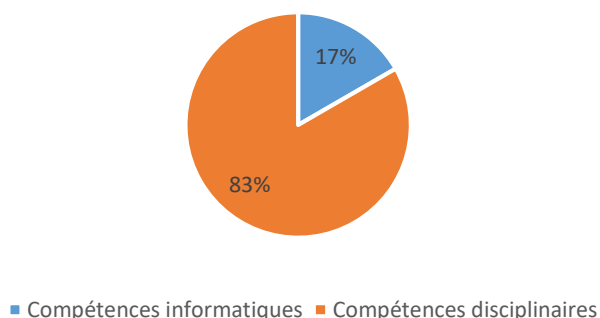
16. Le langage oral a-t-il eu une place importante ?



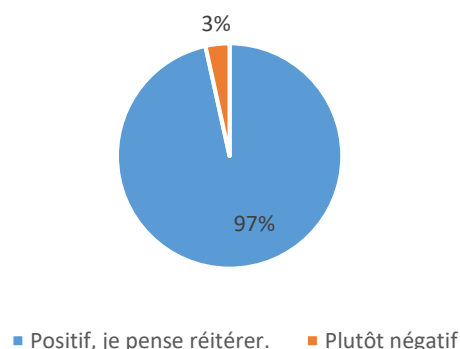
17. Si oui, à quel(s) moment(s) ?



18. Quelles ont été les principales compétences travaillées ?

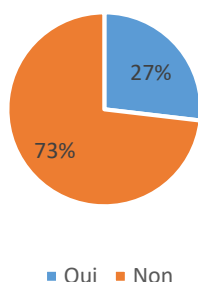


19. Quel est votre retour sur cette séquence ?

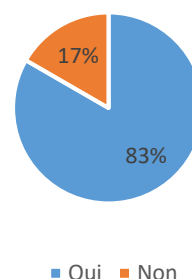


Vous n'avez pas mis en place des séances sur la "programmation"

20. Envisagez-vous de mettre en place des activités liées à la programmation ?



21. Si une formation sur le sujet vous était proposée, l'envisageriez-vous davantage ?



L'enseignement du codage informatique en Europe



ANNEXE 4



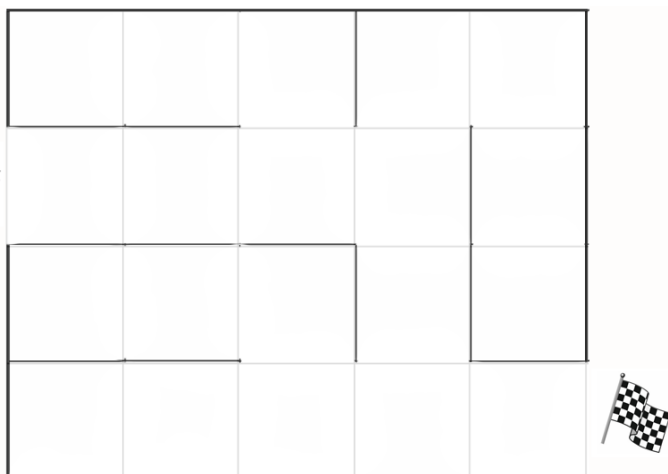
Cadre de référence des compétences

	3.4 Programmer	De quoi s'agit-il ? Écrire des programmes et des algorithmes pour répondre à un besoin (automatiser une tâche répétitive, accomplir des tâches complexes ou chronophages, résoudre un problème logique, etc.) et pour développer un contenu riche (jeu, site web, etc.) avec des environnements de développement informatique simples, des logiciels de planification de tâches, etc... Référence au socle commun : - Savoir que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données (Domaine 1) ; - Connaître les principes de base de l'algorithmique et de la conception des programmes informatiques (Domaine 1) ; - Mettre en œuvre les principes de base de l'algorithmique pour créer des applications simples. (Domaine 1) Thématiques associées : Algorithme et programme; représentation et codage de l'information; complexité; pensée algorithmique et informatique; collecte et exploitation de données massives; intelligence artificielle et robots.			
		Niveau 1	Niveau 2	Niveau 3	Niveau 4 (en cours)
		Lire et construire un algorithme qui comprend des instructions simples.	Réaliser un programme simple.	Développer un programme pour répondre à un problème à partir d'instructions simples d'un langage de programmation ; Modifier un algorithme simple en faisant évoluer ses éléments de programmation ; Mettre au point et exécuter un programme simple commandant un système réel ou un système numérique.	Écrire un algorithme et développer un programme complexe pour répondre à un problème en utilisant un langage de programmation adapté ; Inscrire l'écriture et le développement des programmes dans un travail collaboratif ; Modéliser et simuler des phénomènes physiques, économiques et sociaux.

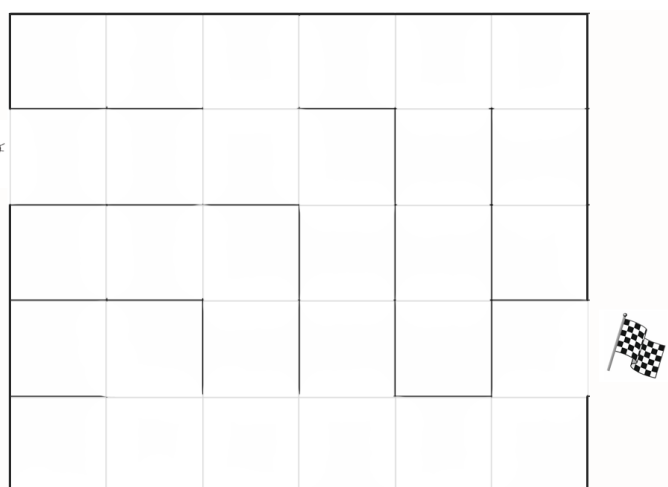
ANNEXE 5

SITUATION DE REFERENCE

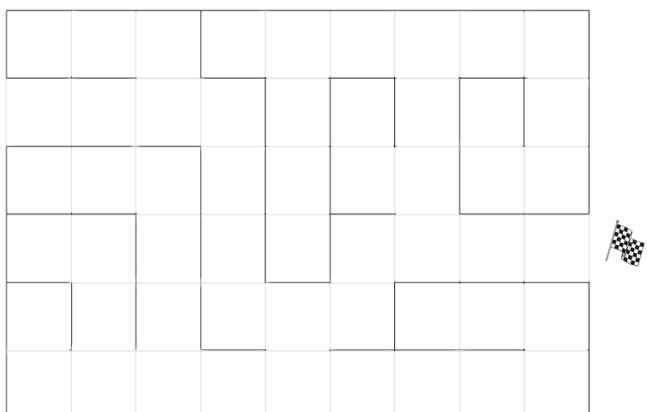
Cycle 1



Cycle 2

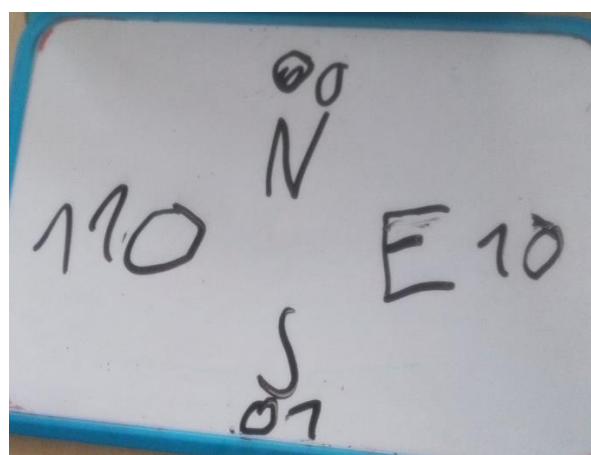
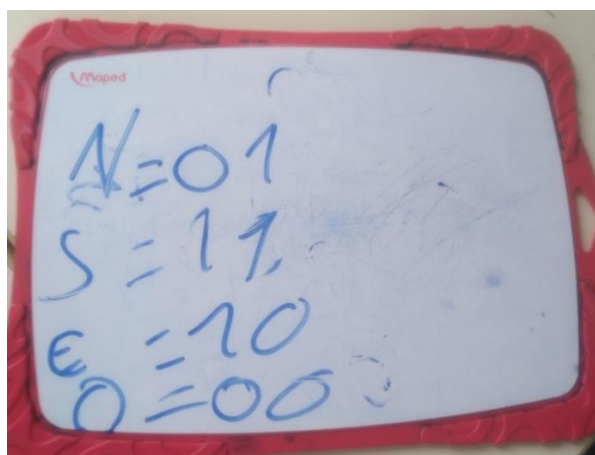


Cycle 3

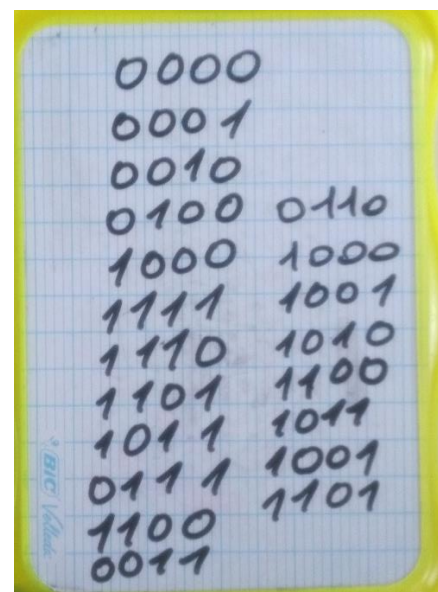
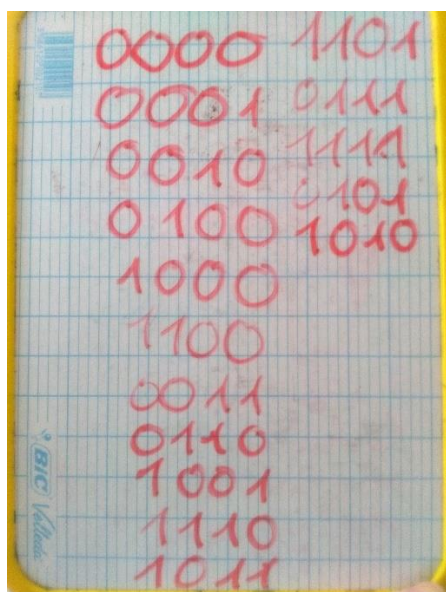


Consigne : Le petit chat "Scratch" veut sortir de ce labyrinthe et attraper le drapeau. Indiquez-lui, de la façon la plus simple possible, le chemin qu'il doit réaliser.

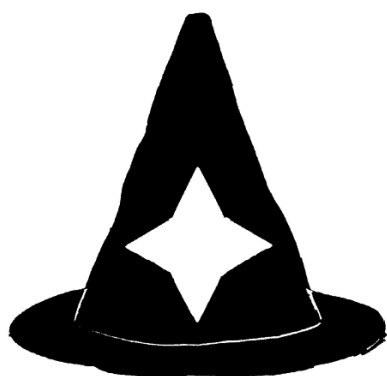
ANNEXE 6



ANNEXE 7



ANNEXE 8



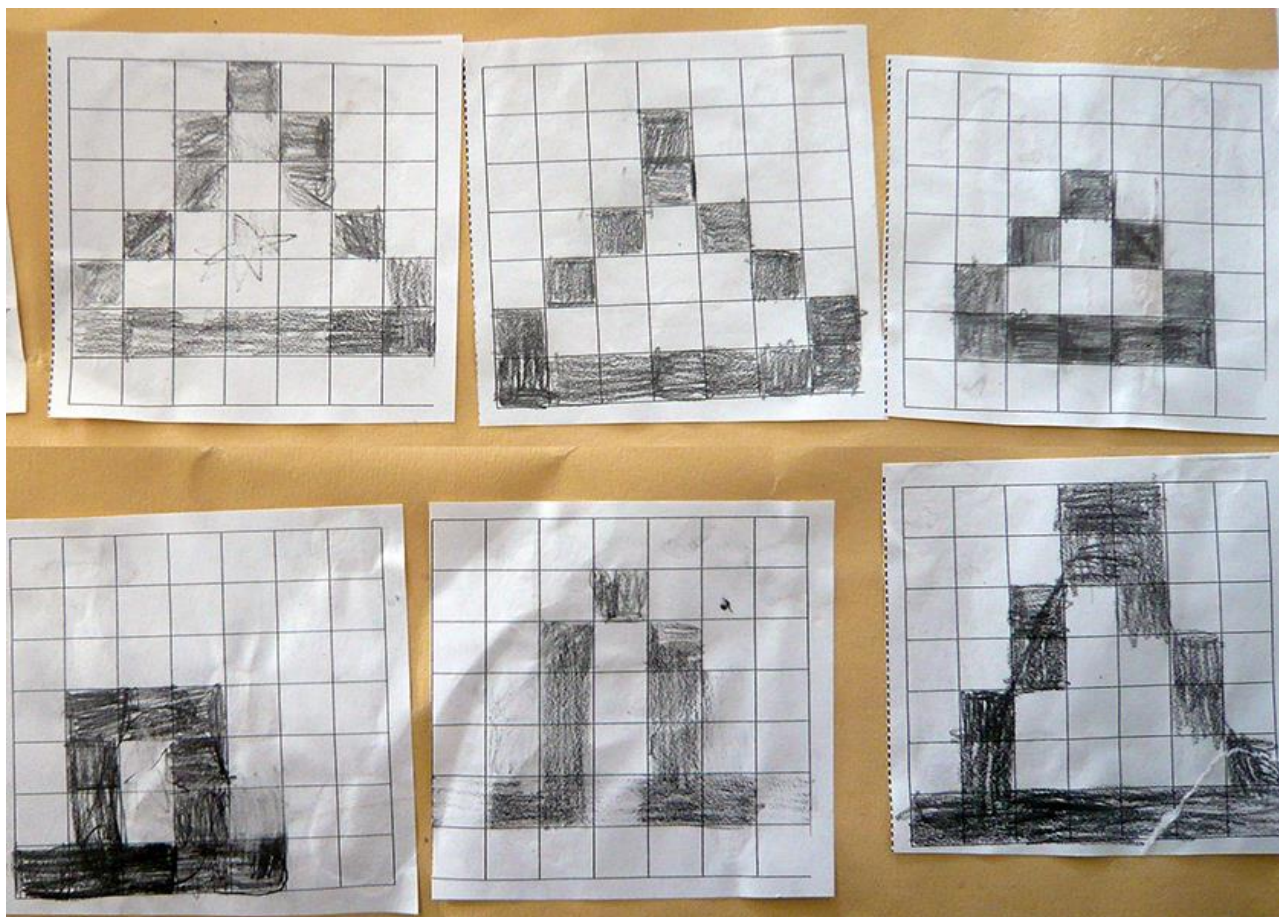
Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



Consigne : Remplis certaines cases de la grille avec du noir pour imiter l'image du chapeau du magicien. Chaque case doit être entièrement blanche ou entièrement noire.



ANNEXE 9

Exercice 1 : faire avancer le chat de 10 pas

avancer de 10

Exercice 2 : faire avancer le chat de 20 pas

Plusieurs solutions possibles :

avancer de 10

avancer de 10

ou

avancer de 20

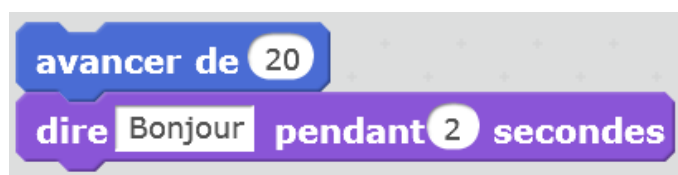
On préférera la seconde solution, plus élégante et facile à lire.

Exercice 3 : remettre le chat au centre de la scène

aller à x: 0 y: 0

Certains élèves vont sans doute trouver l'astuce... mais pour la plupart, il faudra la leur montrer. Malgré tout, il est indispensable pour eux de voir cette instruction dès maintenant, car, à force de déplacer le chat, ils vont le faire sortir de l'écran et ne sauront pas comment le récupérer.

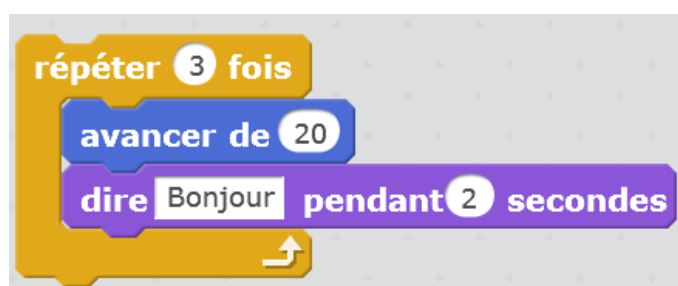
Exercice 4 : faire avancer le chat de 20 pas et lui faire dire « Bonjour »



Bien préciser que « dire » bonjour signifie pour nous « écrire » bonjour : on veut faire afficher une bulle à l'écran avec le texte « bonjour » dedans, on ne veut pas faire parler le chat !

NB : il est possible de faire parler le chat (lui faire émettre des notes de musique ou jouer un fichier son que l'on aura importé ou enregistré dans *Scratch*) ; nous le déconseillons fortement en classe.

Exercice 5 : répéter 3 fois : faire avancer le chat de 20 et lui faire dire « bonjour »



Inciter les élèves qui ne trouvent pas à chercher dans la catégorie « contrôle » (orange). Ils y trouveront une instruction qui ressemble (« répéter 10 fois ») et qu'il est facile de modifier en remplaçant 10 par 3. Cette boucle englobe les autres instructions. Tout ce qui se trouve à l'intérieur de la boucle est exécuté 3 fois.

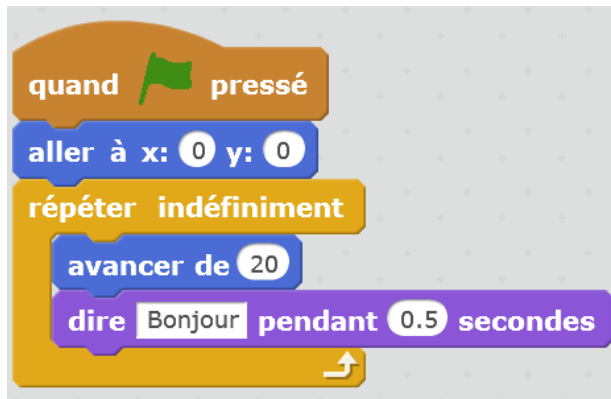
On voit le chat qui s'arrête 2 secondes entre chaque mouvement... pour raccourcir cette pause, il suffit de raccourcir la durée pendant laquelle il dit « bonjour » (si on écrit « 0.5 » au lieu de 2, il ne s'arrêtera qu'1/2 seconde à chaque fois).

Exercice 6 : répéter indéfiniment : faire avancer le chat de 20 et lui faire dire « bonjour »



Cela se fait très simplement, sur le même modèle que l'exercice précédent, mais avec un autre type de boucle.

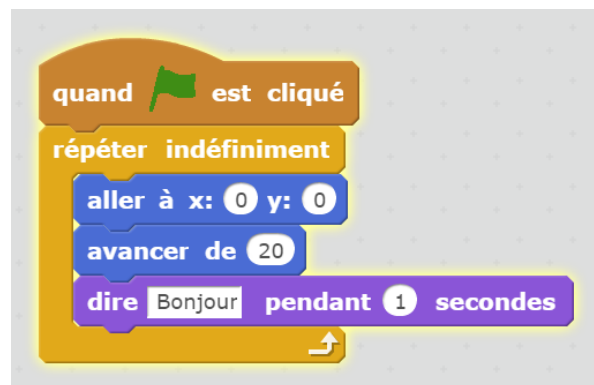
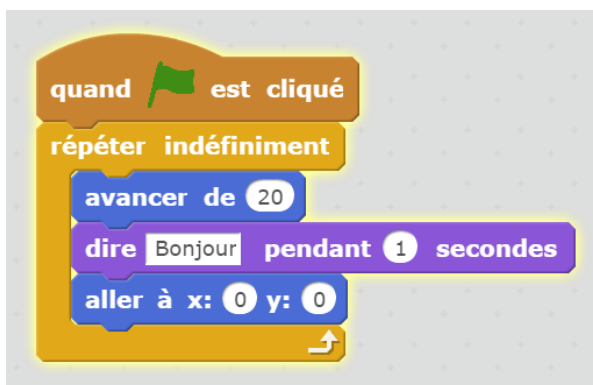
Exercice 7 : même chose quand on clique sur le drapeau vert



Il suffit a priori de rajouter l'instruction « quand drapeau vert pressé » (issue de la catégorie événement), mais cela est encore mieux si on demande au chat de repartir du centre de la scène.

Expliquer, à ce moment, le rôle du bouton rouge (situé à côté du drapeau vert). Un clic sur ce bouton rouge met fin à l'exécution du programme (qui, sinon, ne s'arrête jamais dans le cas présent).

ANNEXE 10



ANNEXE 11

