L'ALGORITHMIQUE ET LA PROGRAMMATION POUR LA CONSTRUCTION DU SENS DE LA DIVISION EUCLIDIENNE

CHAACHOUA[*] Hamid – TCHOUNIKINE[**] Pierre – CRISCI[***] Rosamaria

Résumé – Nous abordons la question de la prise en compte de l'algorithmique à l'école primaire en tant que compétence générale relevant de la résolution de problème. Nous présentons des caractéristiques, au niveau didactique et algorithmique, que doivent avoir les tâches constituantes d'une séquence sur la division euclidienne. En se basant sur ces caractéristiques, l'article présente la construction de la séquence qui fait objet d'expérimentations dans des classes du cycle 3.

Mots-clefs : Algorithmique, Division euclidienne, Variable didactique, Scratch.

Abstract – We address algorithmic as a general problem solving competence at elementary school. We present the didactics and algorithmic characteristics of the tasks constituting a teaching sequence for Euclidean division. We present a teaching sequence based on these characteristics, which is presently under experimentation in third cycle classrooms.

Keywords: Algorithmic, Euclidean division, didactic variable, Scratch

I. INTRODUCTION

L'enseignement de l'informatique est entré dans les programmes scolaires du cycle 3. Outre le fait d'utiliser des logiciels usuels (traitement de texte, etc.), ces programmes indiquent qu'il est souhaitable que les élèves découvrent et pratiquent l'algorithmique et la programmation. Soulignons que l'introduction de l'algorithmique et de la programmation sont au programme du lycée depuis 2010 et certains travaux de recherche en didactique se sont intéressés à l'impact de cette introduction sur l'évolution de l'enseignement des mathématiques (Modeste, 2012) (Haspekian & Nijimbere, 2012).

L'algorithmique est une compétence générale, qui relève de la résolution de problème : un algorithme est un enchaînement d'actions, dans un certain ordre, qui chacune a un effet, et dont l'exécution complète permet de résoudre un problème. Un algorithme se décrit généralement en langage naturel. Il peut faire l'objet d'une programmation, c'est-à-dire d'une traduction dans un langage interprétable et exécutable par un ordinateur. Des langages de programmation pour enfants comme Scratch proposent des structures visuelles (des blocs) et des aides (manipulation des blocs par glissé-déposé, contrôle syntaxique automatique, etc.) qui font qu'il est possible de superposer la phase d'algorithmique et de programmation : les élèves écrivent leur algorithme en agençant les blocs Scratch, ce qui permet de l'exécuter directement.

Dans le cadre du projet EXPIRE[1], nous mettons en œuvre un ensemble d'actions qui vise à coupler une initiation à l'algorithmique et l'enseignement des mathématiques. En particulier,

[*] Univ. Grenoble Alpes, LIG – France – Hamid.Chaachoua@imag.fr

[**] Univ. Grenoble Alpes, LIG – France – Pierre.Tchounikine@imag.fr

[***] Univ. Grenoble Alpes, LIG – France – rosamaria.crisci@univ-grenoble-alpes.fr

[1] EXPIRE (EXpérimenter la Pensée Informatique pour la Réussite des Élèves ; cf. http://lig-membres.imag.fr/tchounikine/projetEXPIRE.html) est une opération soutenue par l'État dans le cadre du volet e-FRAN (Espace de formation, de recherche et d'animation numérique) du Programme d'Investissement d'Avenir, opéré par la Caisse des Dépôts. Il

nous explorons comment utiliser l'introduction en cycle 3 de l'algorithmique et de la programmation pour en faire des vecteurs d'apprentissage de notions de mathématiques.

Dans cet article, nous présentons les principes didactiques et algorithmiques d'une séquence focalisée sur la division Euclidienne. La séquence est fondée sur une situation didactique classique que, par un travail pluridisciplinaire impliquant didacticiens et informaticiens, nous avons retravaillé. Nous montrons comment le fait de demander aux élèves de construire un algorithme et de l'exécuter sur ordinateur peut être utilisé pour la construction du sens des notions de diviseur et de reste.

Le plan de l'article est le suivant : nous présentons d'abord la division euclidienne en tant qu'objet à enseigner à l'école primaire (§ II) et les objectifs de la recherche (§ III). Ensuite, nous présenterons les éléments de constructions de la séquence et de sa mise en place (§ IV et § V) et enfin quelques résultats préliminaires (§ VI).

II. ENSEIGNEMENT DE LA DIVISION EUCLIDIENNE

La division a deux significations qui peuvent être travaillées dans des contextes différents. La première signification est celle de division-partage, que l'on rencontre dans des situations de recherche de la « valeur d'une part ». La deuxième signification est celle de division-quotition, que l'on rencontre dans des situations de recherche du « nombre de parts ».

Plusieurs situations ont été élaborées pour construire le sens de la division euclidienne et en particulier le sens du quotient et du reste. En effet, une des difficultés de cette notion est que c'est une opération dont le résultat est formé de deux nombres.

Dès 1972, Guy Brousseau propose la situation dite de « course à 20 » sous forme d'un jeu entre deux élèves où il s'agit, pour chacun des adversaires, de réussir à dire 20 le premier, en ajoutant 1 ou 2 au nombre dit par l'autre. En fait, cette situation se modélise par le jeu de la « course à n », considérée comme situation fondamentale de la division (Brousseau, 1998). Le développement d'une stratégie gagnante nécessite de calculer le reste d'une division euclidienne.

Différentes ressources pour les professeurs des écoles proposent des séquences pour introduire la division euclidienne. Par exemple, le manuel CAP Maths CM1 (Ermel, 1997) propose une situation autour de la question « combien de fois un nombre est contenu dans un autre ? ». Dans la situation, les élèves doivent trouver le plus grand nombre de rubans identiques, de longueur donnée, dans un ruban donné. Les élèves peuvent manipuler et réaliser des découpages pour répondre à la tâche ou pour vérifier leur réponse.

Au niveau des programmes, la notion de division est abordée au cycle 2, dans des situations simples de partage ou de groupement. C'est au cycle 3 que l'on aborde la division euclidienne de deux entiers, dès le début du cycle.

III. OBJECTIFS DE LA RECHERCHE

Nous avons choisi de travailler le sens de la division euclidienne à partir de la recherche de nombre de parts dans une situation de division-quotition, et plus spécifiquement autour des objectifs suivants :

---

implique l'Université de Grenoble Alpes, la ville de Grenoble, le CCSTI La Casemate, l'Espé et le Rectorat de l'Académie de Grenoble.

- Trouver le nombre de parts dans une situation de partage équitable.

- Comprendre la signification du quotient et du reste.

- Comprendre l'écriture de la division euclidienne

L'algorithmique et la programmation forment une approche de la résolution de problème qui nous semble présenter deux caractéristiques pertinentes. D'une part, écrire un algorithme amène les élèves à expliciter une procédure de calcul. D'autre part, l'algorithme (ou, dit autrement, la procédure de calcul proposée par l'élève) peut être exécuté par l'ordinateur. Le fait que les conditions institutionnelles soient maintenant favorables à la pratique de l'algorithmique et de la programmation en cycle 3 nous a amené à considérer la question de recherche suivante : *comment exploiter les caractéristiques de l'algorithmique et de la programmation pour créer des situations didactiques faisant travailler le sens de la division euclidienne* en cycle 3 *?*

Pour répondre à cette question nous avons considéré l'enjeu d'apprentissage « donner du sens aux notions de quotient et de reste » et construit un ensemble de tâches présentant les caractéristiques suivantes :

1) L'algorithme/programme[2] que l'élève doit écrire définit le comportement d'un objet ou d'un personnage (par exemple, son déplacement) à l'écran.
2) La tâche de l'élève consiste à construire un algorithme qui permet d'obtenir un comportement cible, qui est stipulé par l'énoncé de l'exercice.
3) L'écriture de l'algorithme nécessite la mobilisation des notions enjeux d'apprentissage.
4) L'algorithme attendu est une explicitation de la procédure de calcul attendue.
5) L'exécution de l'algorithme permet de visualiser la procédure proposée par l'élève.
   Notons que cette recherche vise donc à étudier l'apport d'une séquence où algorithme et programmation sont réalisés ensemble et exécutés sur l'ordinateur. Il ne s'agit pas d'étudier l'apport de l'algorithmique en soi, qui peut être travaillée sans ordinateur.

IV. ÉLEMENTS DE CONSTRUCTION DE LA SEQUENCE

Dans la suite, nous présentons les choix fondamentaux de construction de la séquence, qui reposent sur des choix didactiques et/ou des propriétés de l'environnement informatique. Pour cela nous nous référons à la Théorie Anthropologique du Didactique (Chevallard, 1999), et plus précisément au cadre T4TEL (Chaachoua, 2018).

1. Le type de tâches

Il s'agit de considérer une bande numérique avec une *case cible* et un objet qui peut se déplacer sur cette bande en faisant des *sauts* constants.

D	1	2	3	4	…	C

Nous avons choisi comme départ, désigné par D, la case qui précède 1, sans introduire le 0. La case cible est une case désignée par *C*. Pour définir les sauts nous avons introduit une unité *pas*. Ainsi, un saut est définit par un nombre de pas : 1 pas, 2 pas, etc.

[2] Dans notre approche, les élèves écrivent directement l'algorithme dans l'environnement Scratch, et donc écrivent en même temps l'algorithme et le programme exécutable.

Afin de pouvoir travailler sur des nombres qui dépassent 100 et compte tenu des contraintes de l'écran nous avons opté par représenter la bande numérique par une spirale (cf. Figure 1).

Étant donnée une cible C et un saut S inférieur ou égal à C, il existe un couple d'entiers unique (q , r) tels que $C = S \times q + r$.

Les principes de l'environnement de programmation Scratch permettent de mettre en œuvre la situation en répondant aux caractéristiques recherchées (1) et (2). Scratch permet de définir des « lutins » (des objets ou des personnages). Le comportement d'un lutin est régi par un algorithme. L'algorithme s'écrit par composition de blocs, par exemple « avancer de x » (qui fait avancer le lutin sur l'écran). Le lutin apparaît sur une « scène » (un écran), vide par défaut, mais où il est possible de placer un dessin. Ceci permet de créer la situation suivante : une scène représentant une bande numérique ; un lutin, que nous avons décidé de représenter par un cercle-repère, positionné sur D (cf. Figure 1) ; et une tâche relevant du type de tâches T : « écrire un algorithme permettant de faire avancer le cercle-repère de D vers C en respectant un certain nombre de contraintes (taille des sauts, cases à éviter, etc.) ». Les données de la tâche, les contraintes de l'interface à respecter et les propriétés de la mise en œuvre avec Scratch sont liées à différentes variables didactiques[3] qui permettent de générer des types de tâches et des sous-types de tâches à partir de T.

2. Les variables didactiques

Variable V1 : Taille du nombre S (taille des sauts), qui peut prendre deux valeurs :

- Petit, S $\leq$10 (actions possibles : saut de 1 pas, saut de 2 pas, …, saut de 10 pas).
- Grand, S > 10 (actions possibles : saut de 11 pas, saut de 12 pas).

Afin de respecter les critères (3) et (4), nous avons défini des blocs Scratch spécifiques correspondant aux différents sauts autorisés : « Avancer de 1 pas », « Avancer de 2 pas », …, « Avancer de 12 pas ». L'énoncé de l'exercice précise que ce sont ces blocs qui doivent être utilisés pour écrire l'algorithme.

Afin de répondre au critère (5), les blocs sont construits pour marquer un arrêt de quelques dixièmes de seconde à chaque saut. Ainsi, si l'algorithme fait avancer le cercle-repère par sauts de 3 pas, il marque une courte pause en 3, 6, etc.

Variable V2 : Taille du nombre C (valeur de la case cible), qui peut prendre deux valeurs :

- Petit : $C \leq 100$.
- Grand : C > 100.

Les valeurs de S et C permettent de favoriser ou non la mobilisation du répertoire de la multiplication. Ainsi, on s'intéresse à la combinaison de ces deux variables comme suit :

- S $\leq$ 10.
- S > 10 et $C \leq 10 * S$.
- S > 10 et $C > 10 * S$.

Afin de disposer de bandes numériques de longueurs importantes, nous les avons représentées sous la forme de spirales de 85 et 150 cases (cf. Figures 1 & 2). Les blocs

---

[3] Bien que la notion de variable didactique est empruntée à la théorie des situations didactiques nous l'utilisons dans le cadre de la T4TEL (Chaachoua, 2018).

« Avancer de n pas » sont construits pour suivre cette spirale (l’élève n’a pas à « faire tourner » le cercle-repère).

Variable V3 : Reste, qui peut prendre deux valeurs :

- Reste nul.
- Reste non nul.

Dans le cas où le reste est nul, on retrouve les notions de multiples et diviseurs.

Variable V4 : Visualisation des sauts, qui peut prendre deux valeurs :

- Sans trace : chaque saut est indiqué par une courte pause uniquement.
- Avec trace : les cases correspondant aux sauts sont marquées.

Cette variable relève du critère (5), et permet de moduler la visualisation des sauts (i.e., de la procédure de l’élève) pendant l’exécution du programme. Cette rétroaction permet notamment de rendre visible la régularité des sauts, qui est un élément pouvant aider à la compréhension du sens de $q \times S$ dans cette situation.

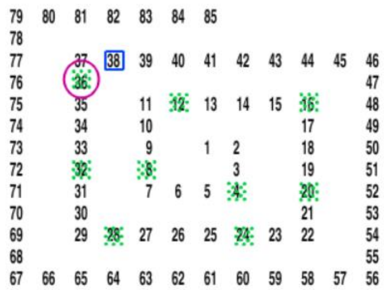

Figure 1 *– La bande numérique (taille 85, cercle-repère en 36, cible en 38, avec traces de sauts de taille 4)*

Variable V5 : Présence de cases interdites

Les cases interdites sont des cases où le cercle-repère ne doit pas s’arrêter. Ce type de contraintes est intéressant dans le cas où le reste est nul et l'élève doit choisir parmi plusieurs sauts possibles.

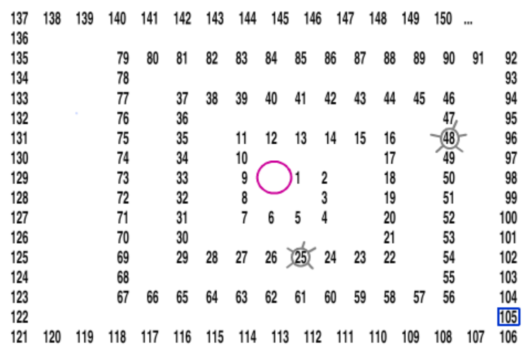

Figure 1 *– La bande numérique (taille 150, cercle-repère en D, cible en 105, avec des cases interdites en 25 et 48)*

Variable V6 : Masquage de la bande numérique

Les valeurs de la bande numérique peuvent être masquées (une série de cases successives sont remplacées par un trait). Cette variable permet de bloquer des procédures basées sur l’énumération.

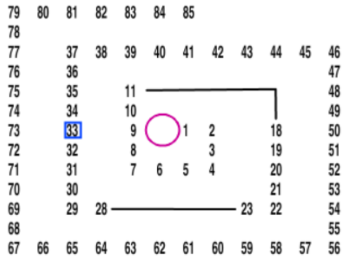

***Figure 1** – La bande numérique (taille 85, cercle-repère en D, cible en 33, avec des cases masquées)*

V. PRÉSENTATION DE LA SÉQUENCE

La séquence est organisée en 4 phases. La première permet de travailler les prérequis sur les multiples et les diviseurs. La deuxième porte sur la détermination du quotient et le sens du reste. La troisième porte sur la détermination du quotient et du reste, ainsi que l'écriture de la division euclidienne. Enfin, la dernière est une phase d'institutionnalisation du savoir appris à travers la résolution du problème.

Nous présentons dans ce paragraphe les choix des valeurs des différentes variables par rapport aux procédures et aux enjeux d'apprentissage dans la construction des trois premières phases. Concrètement, chaque phase correspond à une série d'exercices que les élèves doivent réaliser, chaque exercice correspondant à des valeurs différentes des variables didactiques.

1. Phase 1

L'objectif de cette phase est de travailler les prérequis sur les multiples et les diviseurs. Les tâches stipulent qu'il faut déplacer le cercle-repère sur la bande numérique pour atteindre une case cible en utilisant un type de saut choisi entre 1 et 9.

Considérons 3 tâches correspondantes à des choix de valeurs de variables différentes.

- La tâche 1 où la cible est C = 24. Les sauts possibles sont 1, 2, 3, 4, 6, 8. La technique mobilise la notion de diviseurs, ici les diviseurs de 24.
- La tâche 2, où l'on conserve la cible mais où l'on place des bombes sur 9 et 14. Dans ce cas, seuls les sauts 4, 6 et 8 sont gagnants (il faut choisir un diviseur de 24 qui ne soit pas diviseur des bombes).
- La tâche 3, où la cible est 84 et où l'on place une bombe sur 36. Dans ce cas, il y a un seul saut gagnant : 7.

Dans les deux tâches 1 et 2, le choix de la valeur C = 24 permet la mobilisation du répertoire de multiplication (variable V2). Ensuite, le choix des cases interdites par des bombes (Variable V5) disqualifie certaines techniques basées sur un choix aléatoire du saut. La tâche disqualifie, d'une part, la mobilisation du répertoire de la multiplication, et, d'autre part, le choix aléatoire de saut. Ainsi, la réussite de la tâche 3 nécessite la mobilisation des connaissances sur les multiples et diviseurs.

Nous voyons que ces 3 tâches répondent aux caractéristiques recherchées (1) et (2).

2. Phase 2

Dans cette phase, on introduit un reste non nul (variable V3). Les exercices stipulent qu'il faut que le cercle-repère se rapproche au maximum du nombre encadré C, sans le dépasser, et en utilisant le saut stipulé par l'énoncé de l'exercice.

Plusieurs procédures sont possibles : procéder par tâtonnement, mobiliser des multiples, compter les traces des sauts ou effectuer la division (mais cette dernière elle n'est pas attendue). Nous avons joué sur les valeurs des différentes variables pour faire émerger la division euclidienne.

Dans cette phase, nous avons décidé de garder les traces (variable V4) pour permettre de bien visualiser la régularité des sauts. Celle-ci sert également de contrôle dans certaines procédures. Dans les différents exercices, nous avons combiné les valeurs des variables V1 et V2 par rapport au répertoire de la multiplication.

3. Phase 3

L'objectif de cette phase est l'explicitation du reste. Pour cela, les exercices stipulent qu'il faut utiliser deux types de sauts : un saut (quotient) qu'il peut utiliser autant de fois qu'il le souhaite, et un autre (reste) à n'utiliser qu'une seule fois, et qu'il doit choisir dans une liste.

Exemple d'exercice : Écrivez un programme dans Scratch pour permettre au cercle-repère d'arriver exactement sur le nombre encadré (i.e., la cible, qui est mise en évidence sur la bande numérique), en utilisant le saut indiqué (autant de fois que vous souhaitez), et en utilisant une seule fois un joker au choix (des sauts de 1 à 6). Dans cet exercice, C = 97 et S = 7, et nous avons choisi de ne pas laisser de traces des sauts. L'algorithme attendu est présenté en Figure 4.

Figure 4 – *Un algorithme / programme attendu*

4. Un élément essentiel de chaque phase : la traduction vers le registre numérique

Les séances et exercices que nous élaborées sont fondées sur un choix important : la tâche des élèves est de construire un algorithme, et la rétroaction qu'ils reçoivent est le résultat de son exécution à l'écran. Il n'y a pas volontairement pas de rétroaction automatique sur les caractéristiques de la solution proposée : cette tâche est confiée à l'enseignant.

Le travail de l'enseignant est donc fondamental. En particulier, en plus de l'étayage et/ou de la validation, il est essentiel qu'il fasse des synthèses sur la traduction des algorithmes vers des écritures dans le registre numérique. Ainsi, dans l'exemple de la Figure 4, il faut faire le lien avec l'écriture : $97 = 13 \times 7 + 6$.

Ce type de traduction peut passer par des verbalisations du type "pour atteindre la cible 97 on avance de 7 pas 13 fois, puis on avance de 6 pas". Cette verbalisation peut être traduite sous la forme "Cible = (Nombre de sauts * Type de saut) + Joker" pour amener à l'écriture décontextualisée Dividende = (Quotient * Diviseur) + Reste.

VI. RÉSULTATS PRÉLIMINAIRES

La séquence a été déployée dans une cinquantaine de classes de CM1 et de CM2. Sa mise en œuvre par les enseignants n'a pas révélé de problème.

Les observations que nous avons menées dans certaines classes montrent que les élèves s'impliquent très facilement dans la résolution de problème et la création d'algorithmes. Lors des bilans collectifs, les élèves s'appuyaient sur des notions mathématiques (multiples, diviseurs) pour expliquer l'algorithme qu'ils avaient produit. La traduction de l'algorithme en égalités numériques était un moment fort pour la mobilisation et l'explicitation des connaissances mathématiques.

Nous avons réalisé des pré-tests/post-tests qui ont été soumis aux élèves individuellement dans l'environnement papier/crayon. Chaque test était constitué de 5 types de tâches visant spécifiquement à faire émerger des données relatives à la réponse et à la technique employée. Afin de traiter et analyser ces données, nous avons conçu une grille de codage qui met notamment en évidence les relations entre (1) la réponse de l'élève à l'exercice, (2) la technique utilisée par l'élève et (3) les variables des types de tâches et leurs valeurs : gestion du reste (oui/non), type de situation (division-quotition/division-partage), taille des nombres (petits/grands) et réponse attendue (quotient/reste/quotient et reste). Le traitement exploratoire des données de deux classes de CM2 fait notamment apparaître un taux de variation de +100% sur la validité de la réponse pour les types de tâches nécessitant une gestion du reste. De plus, l'évocation de techniques basées sur l'opération de division posée a également vu une augmentation substantielle (taux de variation égal à +170%). Ce traitement exploratoire suggère que la séquence amène les élèves à travailler les compétences visées.

VII. CONCLUSION

L'enseignement de l'informatique au cycle 3 peut être abordé avec des objectifs très différents, cf. l'analyse présentée dans (Tchounikine, 2016). Dans cet article, nous avons présenté un travail en cours qui étudie comment introduire la pensée informatique à l'école primaire comme un vecteur pour l'apprentissage des mathématiques.

Sur l'exemple de la division Euclidienne, nous avons montré qu'il est possible de créer des tâches où les élèves sont amenés à écrire un algorithme qui mobilise les notions enjeux d'apprentissage mais, également, qui est une explicitation de la procédure de calcul proposé par les élèves. L'exécution du programme permet donc d'obtenir une trace d'exécution de la procédure de calcul que propose l'élève. Cette ressource peut être exploitée par l'élève et, bien sûr, l'enseignant.

REFERENCES

Chaachoua H. (2018) Un cadre de référence pour la formalisation et l'extension du modèle praxéologique. *In 6e congrès pour la Théorie Anthropologique du Didactique*. Autrans. Janvier 2018.

Chevallard Y. (1999) L'analyse des pratiques enseignantes en théorie anthropologique du didactique. *Recherches en didactique de mathématiques*, 19(2), 221-265.

ERMEL (1997). *Manuel Ermel CM1, Apprentissage numériques et résolution de problèmes*. Hatier.

Tchounikine P. (2016) Initier les élèves à la pensée informatique et à la programmation avec Scratch. *Revue EpiNet n° 182*, http://lig-membres.imag.fr/tchounikine/PenseeInformatiqueEcole.html.

Modeste S. (2012) Enseigner l'algorithme pour quoi ? Quelles nouvelles questions pour les mathématiques ? Quels apports pour l'apprentissage de la preuve ? *Thèse de l'Université Joseph Fourier*. Grenoble.

Haspekian M. & Nijimbere C. (2012) Les enseignants face à l'entrée de l'algorithmique dans l'enseignement des mathématiques au lycée scientifique en France. *In Actes CORFEM – 18ème et 19ème colloques (2011 et 2012).* Besançon.