
ENSEIGNER L'INFORMATIQUE A L'ECOLE ELEMENTAIRE AU CYCLE DEUX

Une expérimentation avec Scratch Junior

Nathalie VIGOT
ÉSPÉ de Bretagne

Résumé : Cette expérimentation porte sur l'appropriation du langage de programmation avec l'application Scratch Junior conçue pour les I-Pad avec des élèves de cours élémentaire première année (CE1). Il s'agit d'étudier les stratégies et les procédures utilisées par les élèves lorsqu'ils élaborent un programme pour résoudre une situation-problème. Nous analysons les stratégies, les erreurs et les obstacles rencontrés par les élèves lors de l'élaboration d'un programme dans un projet pluridisciplinaire. Nous nous intéressons aux apprentissages réalisés à l'aide du langage de programmation et notamment aux effets de la programmation sur les connaissances mathématiques.

Introduction

Nous relatons une expérimentation de codage, en classe, avec Scratch Junior¹. Il s'agit d'une classe de cycle 2 de cours élémentaire première année (CE1/second grade) composée de vingt-deux élèves âgés de 7 et 8 ans scolarisés dans le Finistère. Tout au long de l'année scolaire, les élèves se sont familiarisés avec l'outil informatique : le bureau numérique des écoles (BNE), Calcul@tice. Ils découvrent le codage avec l'application Scratch Junior lors d'un projet pluridisciplinaire. Les élèves vont devoir raconter notre sortie en mer à bord du Corentin² pendant la semaine du patrimoine maritime en utilisant un langage de programmation. Nous cherchons à favoriser le retour sur les

connaissances mathématiques par le questionnement et la résolution de défis³. Pour cela, nous avons conçu une situation-problème en lien avec la sortie en mer. Les élèves doivent écrire un programme composé de blocs d'instruction pour supprimer le lutin (le chat), sélectionner un paysage marin, ajouter un bateau, des enfants,

1 Nous remercions très chaleureusement l'École Supérieure du Professorat et de l'Éducation de Bretagne, le site de Quimper pour le prêt des I-Pad.

2 Nous précisons que le Corentin est la copie conforme d'un bateau marchand du XVIIIe et XXe siècle.

3 1, 2, 3 ...Codez ! Enseigner l'informatique à l'école et au collège (cycle 1, 2 et 3). Claire Calmet, Mathieu Hirtzig et David Wilgenbus.

des adultes et coder les déplacements. Nous souhaitons observer comment les élèves utilisent leurs connaissances mathématiques pour s'emparer des défis et de la situation-problème. Nous nous interrogeons sur les effets de la programmation à l'école élémentaire dans les apprentissages mathématiques et notamment pour les élèves moins avancés. Notre analyse prend appui sur de la Théorie de l'Action Conjointe en Didactique (TACD).

La situation-problème

Une situation-problème est pour nous un problème pour lequel l'élève ne dispose pas de solution immédiate. L'élève possède des connaissances anciennes qu'il peut mobiliser pour tenter de répondre à la situation. Les connaissances disponibles ne sont pas toujours suffisantes mais elles permettent à l'élève d'appréhender et de résoudre partiellement le problème. Ensuite, l'élève se trouve devant un obstacle. Il a nécessité de faire évoluer ses connaissances et d'en élaborer de nouvelles. Pour cela, il pratique la stratégie des essais et des erreurs. Les défis proposés permettent de familiariser l'élève avec l'interface de Scratch Junior. Il s'agit de le doter des compétences nécessaires pour appréhender le problème. L'élève possède des connaissances mathématiques sur l'addition, la multiplication, le périmètre du rectangle. En revanche, il ne sait pas coordonner plusieurs sous-programmes. Après la sélection du bateau, des personnages et du paysage (la scène), l'élève devra combiner plusieurs sous-programmes à répéter x fois pour x éléments afin de provoquer le même déplacement. En fonction du point de départ et du déplacement choisi, la variable de la taille des éléments est un nouveau problème. Les contraintes sont découvertes en situation lorsque l'élève élabore de nouvelles connaissances. Le milieu constitué de possibles et de nécessaires représente la situation. Nous

définissons le milieu comme antagoniste puisque les éléments sont à sélectionner par l'élève pour élaborer de nouvelles connaissances. Par milieu, nous entendons également les conditions matérielles, et les connaissances anciennes de l'élève c'est-à-dire les connaissances déjà-là pour permettre l'action. Les décisions et les actions de l'élève constituent le milieu cognitif.

Le contexte et le dispositif de recherche

Il s'agit d'une école primaire française composée de trois classes maternelles et de cinq classes élémentaires. L'école est située près d'une zone d'éducation prioritaire. Elle possède une salle informatique pourvue de 13 ordinateurs. Les élèves de cours élémentaire première année (CE1) s'y rendent une fois par semaine. Nous sélectionnons pour notre problématique les séances défis constitutives du projet pluridisciplinaire avec Scratch Junior. Dans ces séances, les élèves vont devoir mobiliser leurs connaissances mathématiques pour résoudre les défis puis la situation-problème. Nous rappelons que notre problématique concerne les effets de la programmation sur les connaissances mathématiques. En résumé : « le langage de programmation de l'application Scratch Junior pour coder les mouvements permet-il aux élèves de cours élémentaire première année (CE1), et notamment aux élèves moins avancés, de consolider leurs connaissances mathématiques ? ». Nous parlons de l'addition, l'addition répétée, la multiplication, la réduction d'écriture mathématique, la perception de l'espace et les déplacements dans un espace orthonormé.

Nous sommes accueillis à l'École Supérieure du Professorat et de l'Éducation de Bretagne, dans une salle du site de Quimper. Afin de disposer de quinze I-Pad pour l'expérimentation, nous choisissons de faire travailler les élèves

en binôme afin de favoriser les échanges et l'entraide. Cela nous semble également devoir favoriser une meilleure compréhension de l'appropriation de l'application Scratch Junior. Nous précisons que le professeur possède également une tablette de démonstration afin de projeter les manipulations, les démonstrations et les codages réalisés par les élèves. Il nous semble que les temps d'échanges seront ainsi favorisés par le travail en binôme et les temps d'institutionnalisation par les échanges collectifs à partir des démonstrations via l'écran du TBI. Nous cherchons à consolider les apprentissages pour tous les élèves mais également pour les élèves moins avancés qui ne peuvent tirer profit des échanges dans la classe puisqu'ils sont trop éloignés du savoir. Les échanges lors des phases d'institutionnalisation devront rendre compte des connaissances élaborées dans la séance. Dans cet article, le terme « coder » signifie : représenter une information à l'aide de différents symboles. En effet, il s'agit pour les élèves de symboliser un chemin à l'aide de différentes flèches qui indiquent les mouvements d'un personnage. Quant à l'action-conjointe, elle est l'ensemble des relations et des actions qui relie le professeur et l'élève dans le milieu et dans le contrat. Par contrat didactique, nous entendons les habitudes de classe, c'est-à-dire les attendus de l'élève et du professeur par rapport à un savoir donné.

Les objectifs des séances

Les objectifs des séances défis sont en lien avec la connaissance de l'application Scratch Junior, tandis que l'enjeu de la situation-problème est l'utilisation de boucles pour coder les déplacements à l'aide de plusieurs sous-programmes à répéter x fois. Les élèves utilisent leurs connaissances mathématiques déjà-là pour résoudre les défis et la situation-problème. Ils vont être ainsi confrontés à trois défis afin de s'approprier l'interface Scratch Junior.

- Le défi A consiste à faire déplacer le chat de 3 cases « vers la droite » puis de 2 cases « vers le haut ».
- Pour le défi B, il s'agit de faire traverser l'écran au chat de gauche à droite, avec un saut tous les 4 pas. Le départ du chat s'effectue de la colonne 3.
- Quant au défi C, il faut faire faire tout le tour de la scène au chat. Le départ du chat se situe en case (L3, C3) c'est-à-dire ligne 3 et colonne 3.

Les connaissances mathématiques en jeu, dans le défi A, sont la réduction d'une écriture additive et l'usage du nombre. Dans le défi B, les élèves vont questionner leurs connaissances de l'addition réitérée avec la traversée de l'écran. Le défi C montrera comment les élèves usent de la connaissance du rectangle pour répondre à la problématique dans un espace non quadrillé avec le recours possible à la stratégie de l'estimation. Nous précisons que les élèves savent décrire un rectangle et utiliser une règle et/ou une équerre afin de le tracer ou de le reproduire. Ils perçoivent, selon nous, l'égalité des longueurs puisqu'ils envisagent le rectangle comme un carré particulier, c'est-à-dire allongé. La notion d'angle droit n'est pas stabilisée et le périmètre est associé au contour qui définit les limites de la figure plane.

Les savoirs et les savoir-faire

L'usage du nombre, l'addition réitérée, l'estimation d'une distance, le rectangle et le périmètre sont les savoirs mathématiques en jeu dans les séances programmées dans la salle informatique. Il s'agit de séances de réinvestissement qui doivent permettre aux élèves d'user de leurs connaissances mathématiques pour résoudre une problématique dans l'interface de Scratch Junior. Les savoir-faire sont en lien avec l'usa-

ge du numérique et la compréhension de l'organisation du monde informatique.

Analyses et résultats

Le défi A et l'appropriation de l'interface

Le défi consiste à faire déplacer le personnage de trois cases « vers la droite » et de deux cases « vers le haut ». La consigne formulée aux élèves est la suivante : « vous devez faire déplacer le chat de 3 cases « vers la droite » et de 2 cases « vers le haut ». Nous observons deux stratégies en relation avec la connaissance des nombres.

L'élève E1 fait avancer le chat de 3 pas « vers la droite » qu'elle code par l'addition de 3 flèches aller à droite (1 flèche « vers la droite » + 1 flèche « vers la droite » + 1 flèche « vers la droite »). Elle poursuit la programmation de deux pas vers le haut par l'ajout de deux flèches

monter (1 flèche « vers le haut » + 1 flèche « vers le haut »). Les élèves peuvent valider leur proposition en faisant se déplacer le chat. Un binôme découvre alors la présence d'une bulle avec le nombre 1 sous chaque flèche. De plus, une calculatrice apparaît parfois à droite de la tablette. Certains élèves proposent de réduire l'écriture du codage qui comprenait initialement cinq flèches (3 flèches « vers la droite » et 2 flèches « vers le haut »). Le codage comprend maintenant une flèche « 3 vers la droite » et une flèche « 2 vers le haut ». La quantité inscrite dans la bulle indique le nombre de flèches associé au déplacement. Maintenant, le symbole de la flèche indique la direction et le nombre de pas. Les deux codages sont affichés à l'intérieur de la fenêtre de programmation.

Chaque codage réalisé est testé et atteste du même déplacement. Il existe pourtant un obstacle chez certains élèves. Nous faisons l'hypothèse que cette difficulté est en relation

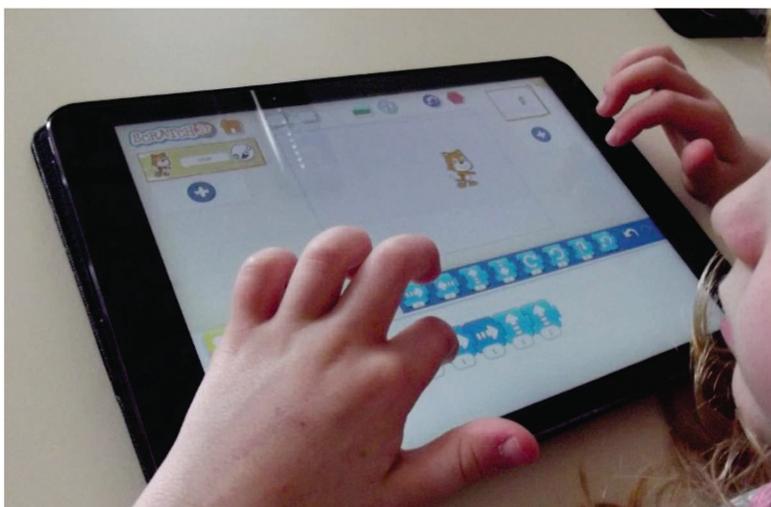


figure 1, le défi A

avec la construction du nombre et peut-être aussi avec la notion de séquence en informatique. En effet, la flèche associée au nombre 3 ne semble pas représenter une quantité égale à trois flèches ajoutées pour les élèves moins avancés. Ils semblent appréhender le nombre principalement par l'unité représentée par la flèche. Les élèves comptent les flèches et sont essentiellement dans le comptage matérialisé par l'ajout successif de flèches.

Le défi B et l'appropriation de l'interface

Le problème consiste à faire traverser l'écran de gauche à droite au chat, avec un saut tous les quatre pas dans un espace plan quadrillé. La consigne donnée aux élèves est la suivante : « vous devez faire traverser l'écran de gauche à droite au chat mais attention le chat réalise un saut tous les 4 pas. Le départ du chat s'effectue de la colonne 3 ». C'est l'occasion de réinvestir l'utilisation du nombre comme quantité pour réduire l'écriture du codage. Ici, nous observons la stratégie de la segmentation du programme de codage dans certains binômes. Quelques élèves cherchent à s'assurer que les instructions codées produisent bien les effets attendus et, pour cela, ils testent le codage du sous-programme « quatre pas et un saut ». D'autres élèves choisissent de coder l'ensemble du programme pour traverser l'écran de gauche à droite avec un saut tous les quatre pas. Certains déplacements sont erronés. Les élèves recherchent les causes des erreurs. Pourquoi le chat se déplace-t-il ainsi ? Comment le déplacement a-t-il été codé pour produire ce résultat ? En fait, des flèches sont graphiquement très proches avec des significations différentes. Les symboles pour tourner « vers la droite », tourner « vers la gauche », « sauter » et « position initiale » sont à l'origine de certaines erreurs de codage. Par exemple, il y a confusion entre la flèche tourner « vers la gauche » et la flèche « position initiale ». De même, nous remarquons des erreurs

entre la flèche tourner « vers la droite » et la flèche « sauter ». La tête des flèches tournent toutes dans le même sens et indiquent bien la même direction d'où la confusion. Pourtant, elles possèdent des différences dans le tracé qui précise le mouvement.

Nous observons également des confusions avec « tourner vers la droite » et « tourner vers la gauche » liées à une latéralisation non consolidée chez quelques élèves. De plus, l'agencement de la salle informatique du site de Quimper à l'École Supérieure du Professorat et de l'Éducation de Bretagne est peut-être à l'origine de certaines confusions. En effet, nous sommes regroupés sur des tables installées au centre de la salle. Les groupes se font face. L'erreur dans le choix des flèches pourrait être dû à une certaine contagion de voisinage. Les élèves observent le codage des élèves placés en face d'eux et ils peuvent vouloir imiter le choix de la flèche. Il existe aussi des erreurs entre les flèches « tourner vers la droite », « sauter », « position initiale » parce qu'elles contiennent toutes une flèche arrondie.



Le symbole de la flèche est utilisé par certains élèves de manière générique au détriment des spécificités non identifiées de chaque symbole. La nécessité d'une activité décrochée pour expérimenter les différentes flèches devrait permettre de surmonter la difficulté observée.

Pour aider à différencier ces flèches, les élèves définissent la flèche « sauter » comme la flèche

« sauter sur place » et la flèche « position initiale » comme revenir au départ (avec la présence de la croix en amont). Les boucles sont bien associées à « tourner », mais ce sont les mots « tourner à droite » et « tourner à gauche » qui sont à l'origine de certaines erreurs.

Le défi C et l'appropriation de l'interface

Le problème consiste à faire faire le tour de la scène par le chat. La scène est représentée par un rectangle. La consigne est la suivante : « Vous devez faire faire le tour de la scène au chat. Le départ s'effectue en bas à gauche (de la scène) ». Les élèves E2 et E3 utilisent la stratégie de l'estimation. Ils utilisent le nombre pour anticiper le nombre de flèches nécessaire pour parcourir la longueur. Ils démarrent avec un nombre estimé avant d'effectuer leur codage afin de pouvoir mieux déterminer si leur réponse est plausible. Ils utilisent les expressions du type « presque, environ ou à peu près ». Pour ce défi, ils ont choisi le nombre 9 pour la flèche « vers la droite ». Ils observent le déplacement et constatent que le chat se retrouve à peu près à la moitié de la longueur. Ils décident de recommencer. Cette fois-ci, ils doublent le nombre choisi précédemment ($9 \times 2 = 18$). Ils constatent qu'une partie des moustaches est cachée. Ils en déduisent que le nombre 18 est juste un peu trop grand. Ils sélectionnent la flèche avancer « vers la droite » et y ajoutent le nombre 17. Ils reproduisent la même stratégie pour la largeur. Ils estiment que la largeur représente environ la moitié de la longueur. Ils proposent le nombre 9 pour la flèche « vers le haut ». Ils remarquent que ce n'est pas assez mais qu'ils sont très proches. Ils ajustent leur codage avec « une flèche de plus ». Ils observent le résultat produit. Ce n'est pas suffisant. Ils tentent à nouveau un ajout. Ils codent la flèche « vers le haut » avec le nombre 11. Les rétroactions du milieu amènent les élèves à rectifier leur codage, développant ainsi une stratégie d'essais-erreurs. Ils observent le dépla-

cement du chat par rapport au problème à résoudre. Il existe des actions de codage davantage liées aux déplacements du chat sur l'écran et moins à l'utilisation du nombre. L'ajustement de la distance parcourue par le chat est réalisé à partir des essais.

Un débat naît autour des moustaches du chat. En effet, pouvons-nous accepter un codage pour lequel les extrémités des moustaches sont trop proches de la scène et n'ont plus tout à fait la même longueur ? Les codages proposés sont très proches et l'argumentation concerne des ajustements de « + 1 » ou « - 1 ». En fait, les élèves discutent sur les codages autour de la question : « pouvons-nous accepter un codage lorsque l'image du chat est un peu rognée ? ».

La programmation des élèves E1 et E4 est beaucoup moins rapide. Ces élèves utilisent la stratégie que l'on peut qualifier « des petits pas ». Ainsi, elles avancent pas à pas (flèche après flèche) et s'assurent du déplacement du chat dans la scène. La fenêtre de programmation étant limitée, elles ne peuvent contrôler la totalité de leur codage. Elles poursuivent avec la même stratégie pour le nombre de la largeur. Au final, elles proposent trois nombres différents pour faire le tour de la scène. Les erreurs semblent liées à des oublis ou des ajouts de flèches comptées. Leurs connaissances mathématiques sur la définition d'un rectangle et notamment la notion de périmètre devrait leur permettre d'interroger leur codage proposé. Pourtant, il n'en est rien. Elles se concentrent sur le comptage de un en un. Nous constatons que le nombre de flèches est exact pour la largeur mais le nombre de flèches est erroné pour la longueur. La stratégie « des petits pas » semble mieux fonctionner avec une distance plus courte.

Il est à noter également que l'on évaluera peut-être mieux à l'œil un nombre de flèches identique si elles sont peu nombreuses. Nous

apprentissages de l'ensemble de la classe. Les groupes qui utilisent « les petits pas » ne proposent pas les mêmes nombres pour toutes les longueurs. Les élèves sont confrontés à un problème. Si les côtés opposés du rectangle ont la même longueur, alors pourquoi les codages réalisés par certains binômes possèdent-ils différents nombres ?

Les groupes qui utilisent « l'estimation » sont momentanément désorientés. Ils ont bien les mêmes nombres pour les longueurs et les largeurs mais leur programme n'indique pas deux fois le même programme. Ils n'ont pas reproduit deux fois le codage du même sous-programme.

Certains élèves hésitent. La longueur et la largeur à reproduire sont à l'origine d'une enquête au sens de la démarche d'investigation. Les connaissances déjà-là sur le rectangle permettent aux élèves d'interroger à nouveau leur programme.

L'élève E5 code le programme avec 17 flèches « vers la droite » et 11 flèches « vers le haut ». Elle reproduit ce sous-programme une seconde fois. Le chat sort de l'écran. Il ne fait pas le tour de la scène comme prévu mais il disparaît de la scène. L'élève E5, étonnée, se demande où est passé le chat ? La classe constate que cela ne fonctionne pas ! Le défi n'est pas réalisé. Pourquoi ? D'où vient l'erreur ?

L'élève E2 répond : « oui... mais il faut changer de direction ». L'élève E2 code le programme suivant : 17 flèches « vers la droite », 11 flèches « vers le haut », ensuite 17 flèches « vers la gauche » puis 11 flèches « vers le bas ». Le défi est réussi même si les élèves n'ont pas utilisé la flèche « tourner ».

En résumé, nous disons que la stratégie de l'estimation permet un gain de temps conséquent.

Elle est liée à la stratégie des « ajustements » et l'emploi de « plus une flèche » ou « moins une flèche ». Néanmoins, la stratégie « des petits pas » apparaît plus facile et plus sûre à certains élèves. En fait, elle est plus longue et à l'origine de nombreuses erreurs. Les connaissances déjà-là de l'élève E5 sont importantes pour consolider les connaissances sur le rectangle. Un rectangle est un parallélogramme particulier dont les côtés opposés ont la même longueur deux à deux. Ce quadrilatère a quatre angles droits et les diagonales de même longueur se coupent en leur milieu. De plus, il possède deux axes de symétrie (les médiatrices de ses côtés) Il est donc juste de formuler que le rectangle est défini par une longueur et une largeur reproduites deux fois. Pourtant, l'élève E5 a délaissé un élément important. Il concerne l'orientation du chat dans le déplacement. Le codage du déplacement lui permet un retour sur ses connaissances mathématiques. Quant aux rétroactions du milieu, elles permettent d'interroger le savoir et de pouvoir en faire usage pour résoudre un problème.

Il pourrait être intéressant de prévoir une séance en informatique décrochée afin d'observer ce qu'aurait fait l'élève E5 qui n'utilise que la fonction avancer. Elle aurait sans doute simplement tourné la vignette du chat tandis que les autres élèves travaillent dans un repère fixe. Nous rappelons que « tourner » correspond à une rotation d'un quart de tour.

Conclusion

Nous nous demandions si le codage avec le langage de programmation à partir de l'application Scratch Junior permettait aux élèves de cours élémentaire première année (CE1) de consolider leurs connaissances mathématiques ?

Pour les élèves E2 et E3, une distance est un nombre que l'on peut estimer et ajuster.

Pour les élèves E1 et E4, la connaissance du nombre est fortement ancrée dans l'unité. Nous voyons qu'elles choisissent de coder au fur et à mesure le déplacement qu'elles suivent sur l'écran. La fenêtre de programmation est limitée. Pour les élèves E1 et E4, la totalité de la programmation n'est pas visible. Nous faisons l'hypothèse que les élèves qui choisissent la stratégie « du pas à pas » sont des élèves encore dans le comptage. Notre hypothèse est construite sur la difficulté pour ces élèves de percevoir et de comprendre que deux codages différents peuvent représenter le même déplacement. Par exemple, lors du défi A, l'objectif est de prendre conscience qu'une même réalité peut être codée par plusieurs programmes. Ainsi une flèche, plus une flèche et encore une flèche sont équivalentes à une flèche associée au nombre 3 à condition que les flèches soient dans la même direction et le même sens. Il s'agit non seulement de réduire le codage du programme apparaissant dans la fenêtre mais plus encore d'utiliser le nombre pour représenter une quantité et/ou une distance dont l'unité correspond à une flèche.

Nous nous interrogeons sur la programmation et le transfert de connaissances. L'élève E5, élève moins avancée, a identifié le rectangle comme représentation de la scène de Scratch. L'élève E5 possède la connaissance mathématique que deux longueurs et deux largeurs « forment » un rectangle. La longueur A est égale à la longueur B comme la largeur A' est égale à la largeur B', ce qui amène l'élève E5 à affirmer que c'est deux fois pareil. Ces connaissances sont à la fois une connaissance déjà-là et peuvent être un obstacle si la connaissance reste figée sans rapport avec les spécificités du problème. Cela peut amener une difficulté dans l'élaboration et la poursuite des apprentissages. Nous pouvons dire que la connaissance du rectangle permet à l'élève E5 de s'emparer du problème et de proposer un coda-

ge qu'elle devra ensuite réinterroger après la réalisation du déplacement du chat.

Le partage des connaissances entre les élèves permet-il la diffusion du savoir ? L'ensemble de la classe est d'accord avec la définition du rectangle proposée par l'élève E5 : « c'est deux fois pareil ». Elle évoque la longueur et la largeur multipliées par deux. Toutefois, si cette connaissance crée un doute chez certains élèves, elle permet aussi de construire une argumentation. Finalement une partie de la classe semble être d'accord avec l'élève E5 sur le codage du sous-programme multiplié par deux fois. Le nombre qui représente la longueur A doit être le même pour la longueur B puisque les deux longueurs sont égales. Pourtant, certains élèves ne sont pas d'accord sur la reproduction du codage du demi-rectangle à l'identique. L'élève E2 argumente que le chat doit changer d'orientation avant de poursuivre son déplacement dans le but d'effectuer le tour de la scène.

L'élève E5 maintient que c'est deux fois pareil. Nous laissons l'élève E5 programmer son codage sur la tablette du professeur connectée au Tableau Blanc Interactif (TBI) et suivre le déplacement. Le chat sort de l'écran après avoir parcouru une longueur et une largeur. L'élève E5 comprend d'où vient le problème. Elle ne prend pas en compte une donnée importante, le changement d'orientation pour résoudre le problème. Les connaissances sur la représentation d'un rectangle sont nécessaires mais elles ne suffisent pas totalement. L'élève doit prendre en compte la donnée de l'orientation dans l'élaboration d'une stratégie experte à partir de savoirs acquis pour résoudre le défi. Le chat doit changer de direction pour parcourir le tour du rectangle c'est-à-dire la scène.

La stratégie experte dépend à la fois de connaissances sur lesquelles s'appuyer mais aussi d'une réorganisation des connaissances déjà-

là en situation. Les connaissances informatiques, la tablette, l'application scratch junior, les connaissances mathématiques participent à la compréhension du micro-espace plan. L'élève E5 modifie ses connaissances en appui sur le milieu (connaissances déjà-là plus milieu matériel) et les échanges au sein de la classe.

En résumé, les connaissances sur les nombres déjà-là (le nombre pour anticiper, le périmètre du rectangle, la réduction d'écritures mathématiques, la multiplication) possédées par les élèves interviennent dans la mise en œuvre d'une stratégie pour coder le programme. La réorganisation de la connaissance déjà-là sur laquelle s'appuyer permet à l'élève de proposer un codage puis de résoudre le problème. Les échanges dans la salle informatique favorisent le partage. Une connaissance réorganisée pour résoudre un problème va pouvoir changer de statut et s'élaborer en savoir lors des phases d'institutionnalisation. L'outil informatique, la tablette pour notre expérimentation, a des effets positifs dans les apprentissages par les essais et les erreurs qu'elle autorise. Les élèves moins avancés construisent la connaissance non plus comme un savoir livresque mais comme une action, un agir qui aide à résoudre le problème. Ils enquêtent ensemble et recherchent la preuve construite sur une argumentation validée par le déplacement effectué.

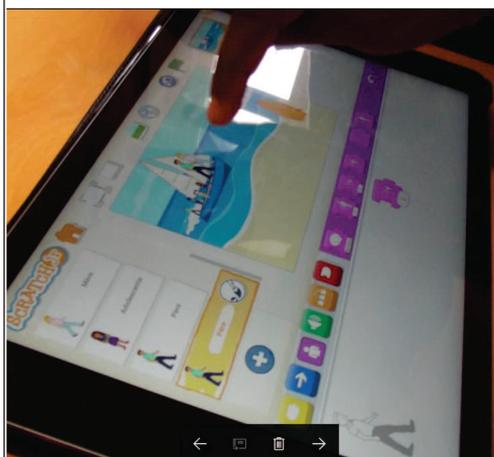
Le travail de groupe constitué de deux élèves (le binôme d'élèves) pour une tablette a généré peu de conflits. Nous avons constaté plutôt un travail d'entraide nécessaire au partage des connaissances et surtout une mutualisation des savoir-faire. Lors des séances, nous avons subi plusieurs déconnexions avec le système de projection. Cela représente, à notre avis, l'écueil majeur rencontré dans la mise en œuvre du dispositif élaboré pour familiariser les élèves au langage de programmation avec Scratch Junior.

Pour conclure, il serait souhaitable, dans le cadre d'une nouvelle expérimentation, de pratiquer une familiarisation avec l'application Scratch Junior à partir des I-Pad sur un temps plus long au début de l'année scolaire. Quant à nous, nous espérons poursuivre l'expérimentation du langage de programmation avec Scratch sur les ordinateurs de l'école. L'expérimentation élaborée ci-dessus nous a livré quelques éléments sur la façon les élèves utilisent leurs connaissances mathématiques pour résoudre des problèmes. La poursuite de l'enquête sur un temps plus long est à envisager afin de pouvoir quantifier et caractériser les effets de la programmation sur l'évolution des connaissances mathématiques pour des élèves de cours élémentaire première année (CE1). Nous terminons notre article sur quelques éléments relatifs à la situation-problème qui consiste à raconter notre sortie en mer. Les élèves sont confrontés à la nécessité d'utiliser les boucles mais également à la taille des personnages embarqués sur le bateau.

Les quelques captures d'écran, ci-dessous, illustrent notre propos. Après avoir sélectionné un paysage marin, les élèves découvrent l'icône qui permet d'ajouter des personnages. Ils sélectionnent des enfants et des adultes qu'ils placent dans le paysage. Ils choisissent ensuite un bateau (figure 3, page ci-contre).

Les images des enfants et les adultes sont beaucoup trop grandes et représentent un nouvel obstacle. Elles ne peuvent être placées sans transformation sur le bateau. Les élèves utilisent la brique « apparence des personnages » déjà utilisée pour faire disparaître le chat pour le rendre invisible. Le problème de la taille des personnages amène les élèves à approfondir leur connaissance des briques « apparence des personnages ». Les élèves activent la brique 2 « rétrécir le personnage ». Nous constatons la même procédure pour tous les groupes. Il s'agit

Figure 3, les éléments importés et le défi de la taille



du choix d'un personnage, son rétrécissement à l'aide de la brique, son positionnement sur le bateau et enfin des ajustements par rapport au bateau dans le paysage. La personnalisation du paysage avec les poissons et les oiseaux de

bord de mer permet aux élèves d'approfondir l'interface graphique de l'application Scratch Junior. Les poissons observés lors de notre embarquement ne sont pas des poissons vivant dans les eaux équatoriales aux flamboyantes couleurs.

La figure 5 (page suivante) représente la production d'un binôme. Nous remarquons que les élèves ont contourné les difficultés rencontrées par l'écriture de sous-programmes pour chaque élément ajouté au décor. Ils effectuent de nombreux essais. Ce sont les rétroactions du milieu qui amènent les ajustements. Après plusieurs essais pour réduire les personnages, les élèves utilisent l'estimation en prenant appui sur les expériences antérieures construites. La réduction de plusieurs personnages amène les élèves à user de l'estimation. Ils font des comparaisons. Ils utilisent alors l'estimation en se basant sur les personnages précédents avant de les placer sur le décor. Les élèves ne semblent pas anticiper le codage ni l'organisation du programme. Ils cherchent à obtenir une réalisation complète. Nous pensons que l'organisation, l'an-

Les briques d'apparence des personnages



Figure 4, le défi du changement de couleur



Figure 5, la sortie en mer

ticipation et la planification d'opérations complexes demandent du temps. Les trois séances prévues pour familiariser les élèves au langage de programmation ne sont pas suffisantes pour atteindre cet objectif. Dans une séance supplémentaire, les élèves se sont essayés à faire parler les personnages pour raconter notre sortie en mer sur le Corentin.

Au cours de ce défi, les élèves se sont familiarisés avec de nouvelles fonctions. Ils ont utilisé l'icône « écrire du texte » et les briques de son pour enregistrer des bruits ou faire parler les personnages mais là encore, le temps a manqué.

Pour l'utilisation des boucles, les élèves pensaient unanimement pouvoir contrôler le déplacement du bateau avec les personnages par un seul programme. Ils constatent que le déplacement codé n'est pas du tout réaliste. Les adultes et les enfants restent posés à la surface de l'eau et le bateau effectue, seul, son déplacement. Nous travaillons à imposer une position initiale afin de pouvoir procéder par essais-erreurs avec l'ajout de l'instruction « aller à x :...aller y :... ». Le terme de boucle n'a pas semblé poser de difficultés aux élèves. Ils entourent les instructions dans un étai à répéter x fois mais le temps prévu n'a pas permis de finaliser la situation-problème.

Bibliographie

- Baron, G.-L., & Bruillard, E. (2001). Une didactique de l'informatique ? *Revue française de pédagogie*, 135, 163-172.
- Baron, G.-L., & Voulgre, E. (2014). Initier à la programmation des étudiants de master de sciences de l'éducation ? Un compte rendu d'expérience. Drot-Delange, B. ; Baron, G.-L. & Bruillard, E. In *Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif*, 2013, Clermont-Ferrand, France. . Actes du quatrième colloque international DIDAPRO 4-Dida&Stic, 24-26 octobre 2011, Université de Patras. (p. 271-281).
- Béziat, J. (2008). Les TIC et l'école primaire. Le cas français : 1976-2002. http://www.adjectif.net/spip/IMG/pdf/TIC_et_primaire.pdf
- Brown, A. L. (1992). Design Experiments: Theoretical and Methodological Challenges in Creating Complex Interventions in Classroom Settings. *The Journal of the Learning Sciences*, 2, 2, 141- 178.
- Clements, D. H., & Nastasi, B. K. (1999). Metacognition, learning, and educational computer environments. *Information Technology in Childhood Education Annual*, 1, 5-38.
- Demo, G. B., Marciano, G., & Siega, S. (2008). Concrete Programming: Using Small Robots in Primary Schools. In *Advanced Learning Technologies: Proceedings of the Eighth IEEE International Conference on Advanced Learning Technologies* (pp. 301-302).
- Denis, B., & Baron, G.L. (1994). Regards sur la robotique pédagogique. In *actes du quatrième colloque international sur la robotique pédagogique*. Paris : INRP.
- Greff, É. (1998). Le « jeu de l'enfant-robot » : une démarche et une réflexion en vue du développement de la pensée algorithmique chez les très jeunes enfants. *Revue Sciences et Techniques Educatives*, 5(1), 47-61.
- Greff, E. (1999). En quoi le robot Algor constitue-t-il un objet didactique original ? *La revue de l'EPI*, 93, 127-150.
- Greff, E. (2001). Résolution de problèmes en grande section autour des pivotements à l'aide du robot de plancher. *Grand N*, 68, 7-16.
- Highfield, K., Mulligan, J., & Hedberg, J. (2008). Early mathematics learning through exploration with programmable toys. *Proceedings of the joint meeting of PME 32 and PME-NA XXX*, vol. 3 (pp. 169–176), Morelia, México, July 17–21, 2008.
- Komis, V., & Misirli, A. (2011). Robotique pédagogique et concepts préliminaires de la programmation à l'école maternelle: une étude de cas basée sur le jouet programmable Bee-Bot. In *Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques*. Actes du quatrième colloque international DIDAPRO 4-Dida&Stic, 24-26 octobre 2011, Université de Patras. (p. 271-281).

- Komis, V., Misirli, A., (2015). Apprendre à programmer à l' école maternelle à l' aide de jouets programmables in GeorgesLouis Baron, EricBruillard et Béatrice Drot-Delange (dir.) Informatique en éducation: perspectives curriculaires et didactiques, pp. 210-226, Clermont-Ferrand: Presses Universitaires Blaise-Pascal. ISBN :978-2-84516-697-4
- Pekarova, J. (2008). Using a programmable toyatpreschoolage: Why and how?Proceedings Workshop of SIMPAR 2008 International Conference on Simulation, Modeling and Programming forAutonomous Robots, 112-121.
- Poullain, L., Girre, E. &Arrieta, M. (2015). Osons la robotique pédagogique en primaire. Bulletin de l'Unions des Professeurs de Physique et de Chimie, vol. 109, 383-394.
- Sensevy, G & Mercier, A (2007). Agir ensemble : l'action didactique conjointe du professeur et des élèves. Presses Universitaires Rennes (PUR)
- Sandoval, W. A. & Bell, P. (2004). Design-BasedResearchMethods for Studying Learning in Context: Introduction. EducationalPsychologist, 39, 4, 199- 201.