

La programmation par passage des messages dès l'école maternelle ? Le cas de ScratchJr.

Sevastiani Touloupaki & Georges-Louis Baron

¹Laboratoire EDA, Université Paris Descartes, 45 rue saints-pères, 75006 Paris, France

sevina.touloupaki@gmail.com
georges-
louis.baron@parisdescartes.fr

Abstract. Le travail qui suit s'inscrit dans le cadre d'une thèse élaborée en cotutelle entre l'Université Paris Descartes en France et l'Université de Patras en Grèce, portant sur l'apprentissage de l'informatique par de très jeunes enfants. L'objet de cette étude a été l'analyse de la manière dont les élèves de l'école maternelle peuvent mettre en œuvre des concepts de programmation à l'aide du logiciel ScratchJr. Dans ce contexte, nous avons réalisé une étude exploratoire dans une école maternelle grecque auprès de douze élèves de grande section. Pour les besoins de cet article nous avons choisi de nous concentrer sur l'analyse du concept de la communication par passage de messages. Les résultats présentés proviennent de l'analyse des programmes développés par les élèves pendant la séance d'évaluation. Ils soutiennent l'idée qu'il est possible, par la mise en œuvre d'un scénario pédagogique spécifique, d'utiliser dès un très jeune âge la communication par passage de messages, pour synchroniser les actions de personnages différents.

Keywords: Didactique de programmation, Apprentissage, ScratchJr, Message, École maternelle, Programmation visuelle.

1 Introduction

Depuis quelques années, il existe un mouvement international autour de l'éducation à l'informatique, et en particulier à la programmation, pour tous, depuis même la petite enfance. Plusieurs pays d'Europe ont ainsi introduit des enseignements d'informatique et de programmation en primaire ou en secondaire. À partir d'octobre 2016, les nouveaux programmes scolaires en France ont comporté une introduction de la robotique et de la programmation dès l'école élémentaire. Ce choix provient d'une volonté de développer chez les élèves des compétences en programmation, afin de les préparer à devenir des acteurs du monde numérique qui les entoure.

Des recherches ont été menées depuis longtemps sur l'apprentissage de la programmation par de très jeunes enfants, en particulier en Logo (Robert, 1985). Plus récemment, il a été confirmé que les jeunes enfants peuvent réaliser des programmes dès l'âge

de 4 ans (Bers, 2012). Fessakis, Gouli et Mavroudi (2013) ont montré qu'à l'aide d'un environnement de programmation de type Logo, des activités et du matériel approprié, les enfants de 5 à 6 ans peuvent être familiarisés avec des concepts de base de programmation et développer deux stratégies différentes de résolution de problème. Un peu plus tard, Misirli et Komis (2016), décrivent dans un ouvrage les résultats de l'expérimentation d'un scénario pédagogique fondé sur l'usage des jouets programmables Bee-Bot, pour enseigner aux élèves des notions de programmation tels que la séquence, le programme, la commande etc. Cette recherche a été menée dans 34 classes de maternelle en Grèce, auprès des très jeunes élèves de 4 à 6 ans.

L'apprentissage de l'environnement ScratchJr a suscité un nouvel intérêt pour l'apprentissage de rudiments de programmation dès la petite enfance. Ce système met en œuvre des éléments de différents paradigmes de programmation, tels que la programmation structurée, événementielle et orienté-objet. Il s'agit d'un langage visuel et graphique, destiné aux enfants dès 5 à 7 ans, qui a été créé afin de permettre aux enfants d'apprendre à programmer pendant qu'ils créent leurs animations (Flannery, Kazakoff, Bonta, Silverman, Bers & Resnick, 2013).

Nous allons nous intéresser ici à une fonctionnalité peu étudiée au niveau de la petite enfance : la communication par passage de messages. Ce dernier nous rappelle le modèle de « passage de message synchrone » ou « synchronous message passing » en anglais, qu'on retrouve dans des systèmes concurrents (Ben-Ari, 1996). ScratchJr met en effet en œuvre des éléments de programmation concurrente, car il permet de lancer en même temps aussi bien les scripts de deux personnages, que les scripts d'un même personnage, à l'aide de l'événement « Quand drapeau vert appuyé ». La communication entre entités est implémentée par deux commandes sur ScratchJr : la commande « envoyer un message », qui envoie un message de couleur spécifique à un personnage et la commande de « quand message reçu », avec laquelle le script du personnage, dont la commande du message a cette couleur, est déclenché.

S'agissant de l'appropriation de la notion de message au niveau de la scolarité obligatoire, nous pouvons mentionner ici des travaux menés à l'aide du logiciel Scratch.

Meerbaum-Salant, Armoni et Ben-Ari (2010), présentent une étude exploratoire, auprès de deux classes de 3^e (dix-huit et vingt-huit élèves), dont l'objectif a été d'évaluer la capacité pour les élèves à apprendre des concepts d'informatique. Pour cela, les auteurs ont formé les deux enseignants sur le processus de développement d'un programme, afin de leur permettre de mettre en place l'enseignement des concepts informatiques avec Scratch. Il est important de noter ici que les sujets de l'étude n'avaient aucune expérience préalable avec le logiciel Scratch. Les résultats de cette étude montrent que le taux de réussite au post-test était le deuxième meilleur après celui de boucle conditionnelle. En effet, 62% des participants sont arrivés à donner une définition correcte du concept des messages au post-test, tandis que le taux de réussite pour le concept d'initialisation était d'environ 20% et moins de 10% pour celui de variable.

Par la suite, dans une étude hors du contexte scolaire, pendant un camp des vacances d'une durée de deux semaines aux États-Unis, vingt-deux élèves de collège sans expérience préalable sur Scratch ont été évalués en fonction de leur performance aux concepts informatiques présentés. À la fin de cette expérience, les élèves ont pu développer

des compétences en programmation événementielle mettant en œuvre l’envoi et la réception de messages.

Plus précisément, la majorité des participants a utilisé la communication par messages, sauf un groupe qui a eu besoin d’aide et un autre qui a eu des problèmes avec le temps. Pourtant, les résultats montrent aussi qu’ils existent souvent des projets incomplets, où l’une de deux commandes de messages manque, soit parce que les participants ont oublié d’effacer les commandes non nécessaires de l’esce de programmation, soit parce qu’ils n’ont pas terminé leurs projets (Franklin, Konrad, Boe, Nilsen, Hill, Lern, Dreschler, ...Suarez, Waite, 2013).

Dans un article plus récent, Fatourou, Zygouris, Loukopoulos et Stamoulis (2018), analysent l’apprentissage des concepts de programmation concurrente par des élèves de CM2, sans expérience préalable en programmation et de 6e, ayant un peu d’expérience en programmation. Cent vingt-trois élèves au total, de sept classes d’école primaire en Grèce ont participé à l’étude. Les résultats de cette étude montrent que le passage des messages a posé la plupart de problèmes aux élèves, car seulement un élève sur cinq a réussi de l’utiliser correctement sur son projet final. En effet, comme on pourrait le supposer, les élèves qui arrivent mieux à implémenter des messages sur leurs projets sont ceux de 6e.

1.1 La communication par passage des messages sur ScratchJr

Programmer avec ScratchJr consiste à créer des animations en choisissant des personnages et en rassemblant les commandes appropriées à l’ordre séquentiel correct, afin de faire réaliser aux personnages les actions souhaitées. (Komis, Touloupaki et Baron, 2017).

Comme on l’a vu plus haut, ScratchJr met en œuvre des éléments de programmation concurrente et événementielle. Lorsqu’un projet est lancé, les scripts des personnages qui ont l’instruction « Quand drapeau vert appuyé » au début de leur script, seront déclenchés en même temps. L’instruction « Quand drapeau vert appuyé » correspond à un événement qui est lancé lorsque le « drapeau vert global » est appuyé (Fig. 1).

ScratchJr permet également de contrôler le temps d’exécution des personnages. La communication entre personnages se fait grâce à l’envoi et la réception des messages. Pour cela nous avons besoin de deux commandes : la commande « Envoyer message », et la commande « Quand message reçu » (Fig. 2).

La commande « Envoyer message » se trouve dans la catégorie des blocs de déclenchements ou *triggering* blocs en anglais. Son seul paramètre est la couleur. C’est une instruction de contrôle, dont la forme est celle d’un bloc ordinaire. Elle peut être ajoutée à n’importe quel endroit dans une séquence des commandes. Sa position signale le moment de l’envoi du message par le personnage expéditeur. Après son exécution le personnage expéditeur suspend son script jusqu’à ce que celui du personnage destinataire soit terminé.

S’agissant de la réception de messages, elle se réalise à l’aide de la commande « Quand message reçu ». Il s’agit d’une commande, dont la forme nous rappelle celle de la commande « Quand drapeau vert appuyé » et elle ne se place qu’au début d’un

script. Lorsque la réception d'un message se produit, le programme qui suit la commande « Quand message reçu » s'exécute, sous la condition que les deux commandes sont de la même couleur.

Les commandes de messages permettent de définir l'ordre de l'exécution des scripts des personnages et les faire communiquer.



Fig. 1. Instruction « Drapeau vert » ou « Start on green flag » en anglais et icône « Drapeau vert global » ou « Green flag » en anglais.



Fig. 2. Les instructions des messages.

2 Problématique et questionnement de recherche

Dans un article précédent (Touloupaki et al., 2018) nous avons choisi d'étudier la manière dont les élèves de cours préparatoire arrivent à mettre en œuvre des messages à l'aide du logiciel ScratchJr. Pour le présent travail, nous avons choisi d'étudier la même question, mais auprès d'élèves plus jeunes, ceux de grande section de l'école maternelle.

Pour cette étude nous allons essayer de répondre au questionnement qui suit : Comment les élèves arrivent-ils à utiliser les commandes de messages de manière opérationnelle afin de pouvoir résoudre un problème d'évaluation ?

3 Méthodologie

Pour réaliser cette étude exploratoire nous avons choisi d'effectuer une recherche de conception (*design-based research*). Le scénario pédagogique (le design) est central dans cette approche, visant à évaluer sa capacité à transmettre les savoirs choisis par

son créateur. La recherche de conception permet une compréhension profonde du processus de l'apprentissage (Cobb, Confrey, DiSessa, Lehrer, & Shauble, 2003).

Selon Cobb et al. (2003), un des objectifs principaux de cette approche est d'améliorer le scénario du début à travers des évaluations itératives des différentes conjectures, en faisant en même temps des analyses autour du raisonnement des élèves et de l'environnement d'apprentissage.

À cette fin, notre scénario pédagogique a été modifié plusieurs fois pendant la mise en place du terrain, pour s'adapter aux besoins du public visé. Il a été de nouveau modifié à la fin de la première année de notre terrain, afin de correspondre aux besoins de notre terrain en France, l'année d'après. À noter ici que, pour pouvoir étudier la communication par passage des messages, nous sommes passés par un scénario pédagogique de onze séances.

3.1 Le scénario pédagogique

Pour concevoir le scénario pédagogique de cette étude, nous avons commencé par une analyse du langage de programmation ScratchJr, en termes de fonctionnalités. Ensuite, nous avons considéré les spécificités du groupe d'âge visé, c'est-à-dire le développement cognitif, social et affectif de ces élèves. Notre scénario pédagogique comprend trois types de séances, les séances d'initiation au logiciel et à la programmation, les séances de l'enseignement du contenu choisi et de mise en pratique et les séances de l'évaluation de la performance des élèves.

L'objectif de chaque séance d'enseignement était la production d'une animation. Le scénario a été réalisé en classe par nous-mêmes avec l'accord de l'enseignante de la classe. L'écran de notre tablette a été projeté à l'aide d'un mini vidéo-projecteur, pour que les élèves puissent mieux voir ce qui se passe dans notre projet, sur ScratchJr. L'étude a duré trois mois et chaque séance a duré une quarantaine de minutes.

Dans un premier temps, nous avons réalisé un entretien individuel (pré-test), pour détecter les idées initiales des élèves à propos de l'interface et des commandes étudiées. Nous avons aussi réalisé une activité de familiarisation avec la tablette, l'interface de ScratchJr et quelques principaux concepts de programmation.

Ensuite, nous sommes passés aux séances d'enseignement, pendant lesquelles nous avons présenté les commandes de mouvement, l'exécution séquentielle des commandes, l'initialisation de la position des personnages dans la scène, le déclenchement concurrent des scripts de deux personnages, la répétition d'une séquence des commandes un nombre des fois prédéfini et enfin, les messages. Neuf séances d'enseignement ont été réalisées au total et deux séances de ce scénario ont été consacrées à la communication par passage de messages.

Nous nous sommes fondés sur les observations de l'étude précédente (Touloupaki et al., 2018) et nous avons présenté des projets déjà prêts, contenant les commandes des messages et nous les avons laissés expérimenter et découvrir la fonctionnalité de chacune des commandes, à l'aide de nos questions. Nous avons choisi d'employer deux styles d'enseignement : l'expert et le facilitateur (Grasha, 1994). Selon Grasha (1994), le style d'expert, est focalisé à la manifestation des connaissances et à encourager les élèves à améliorer leurs compétences. Le style de facilitateur est centré aux besoins du

public visé et aux objectifs à atteindre. Il est caractérisé par une adaptation continue aux besoins du public, pour réussir l'objectif de l'enseignement. Strawhacker, Lee et Bers, (2017) montrent que la mise en œuvre de ces deux styles d'enseignement est corrélative d'une haute performance en programmation chez les élèves, sur l'environnement ScratchJr.

Pour faciliter la compréhension des élèves, nous avons choisi d'imprimer les images des commandes du logiciel. Nous avons donc construit les commandes du ScratchJr en version papier, que les élèves ont utilisées pour créer et observer en même temps les scripts de deux personnages communiquant par messages. Ensuite, nous sommes passés aux séances d'évaluation de la performance des élèves.

En effet, nous avons mené, un entretien individuel (post- test) et deux problèmes à résoudre pour tous les élèves. Le premier problème concernait entre autres la notion de messages. Pour résoudre le problème d'évaluation, les élèves ont été invités à regarder une animation en mode de présentation, accompagnée de la consigne du problème donné lue par nous-mêmes, parce que les élèves de l'école maternelle ne sont pas en mesure de lire, et reproduire le même projet final. Les élèves avaient la possibilité de revoir l'animation en plein-écran et de réécouter la consigne du problème s'ils avaient besoin.

Comme évaluation, on demande aux enfants d'observer le programme donné, d'identifier les commandes appropriées et de les utiliser dans l'ordre correct afin de reproduire les comportements observés, pour les personnages participant au projet. L'objectif de cette activité d'évaluation est d'évaluer aussi la capacité des élèves à réaliser ce qu'on appelle « goal-oriented programming ».

3.2 L'échantillon

Cette étude pilote a été menée en 2016-2017 dans une école maternelle publique de la ville de Patras en Grèce, auprès de 12 élèves, 6 filles et 6 garçons. Ces élèves (de 5 à 6 ans) ont été choisis en fonction de leurs réponses au pré-test et à l'aide de l'enseignante de la classe. Ils ont travaillé par groupe de quatre, avec quatre tablettes, pour réaliser les activités proposées par le scénario pédagogique élaboré à ce titre. Nous avons travaillé avec trois groupes de quatre élèves. Les groupes ont été constitués de manière à pouvoir représenter tous les niveaux de compétences scolaires différents présents dans la classe.

Notre échantillon est donc de faible taille, mais il est contrasté, ce qui nous a semblé suffisant pour notre objectif de repérer des faits intéressants qui devront être confirmés ou infirmés par des recherches ultérieures.

3.3 Techniques et outils de recueil des données

Pour recueillir nos données nous avons utilisé comme techniques l'observation participante et les entretiens semi-directifs. Pour faciliter le recueil de données nous avons utilisé un enregistrement audio-visuel du processus, ainsi que nos notes personnelles, en nous centrant sur les incidents critiques, c'est-à-dire les moments importants

pour l'élaboration de la solution du problème donné par les élèves. Nous avons également eu recours à des captures d'écran des programmes élaborés par les élèves tout au long du processus.

4 Résultats

L'analyse des programmes élaborés par les élèves face à un même problème d'évaluation a été conduite en fonction de deux critères : d'abord, s'ils ont utilisé les deux commandes de messages pour synchroniser les actions de deux personnages et ensuite, s'ils ont utilisé les commandes des messages selon l'ordre correct dans le script des personnages concernés.

L'ordre est dépendant de la consigne du problème donné. Dans le Tableau 1 nous présentons l'ordre correct des commandes des messages, selon le problème donné. Il est important de signaler ici que nous avons choisi de ne pas introduire le paramétrage des commandes des messages, c'est-à-dire la couleur.

Tableau 1. Ordre correct des commandes des messages.

Commande	Personnage	Ordre
Envoyer un message	Enfant	La commande se trouve après la fin du déplacement et avant la fin rouge.
Quand message reçu	Papillon	La commande se trouve au début du programme.

Ainsi, nous sommes arrivés à distinguer trois types de performance des élèves : une « utilisation opérationnelle », une « utilisation partielle », et « pas d'utilisation ». Plus précisément, 6 sur 12 élèves arrivent à une « utilisation opérationnelle » des messages (Fig. 3). Ces élèves utilisent les deux commandes des messages, dans l'ordre correct, sur les scripts des personnages demandés. Nous pouvons aussi remarquer que 3 sur 12 arrivent à une « utilisation partielle » des messages, car ils utilisent les deux commandes des messages, mais seulement l'une de deux commandes est à la bonne place (Fig. 4). Enfin, nous pouvons observer que 3 sur 12 élèves n'utilisent pas du tout les commandes des messages.



Fig. 2. Utilisation opérationnelle des commandes des messages.

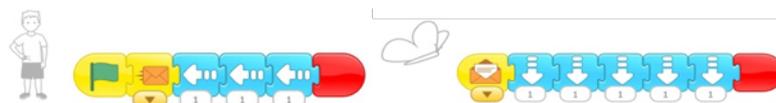


Fig. 3. Utilisation partielle des commandes des messages.

Comme nous pouvons l'observer auparavant la moitié de notre échantillon (6/12) est arrivé à une « utilisation opérationnelle » du concept des messages. Il est intéressant ici de signaler que 5 sur 6 sont des filles. Nous pouvons alors constater que dans le contexte de cette classe et avec ce scénario pédagogique, ces dernières réussissent mieux que les garçons. Ce constat ne permet cependant pas de tirer de conclusions significatives car il pourrait être lié à des niveaux de développement différents entre filles et garçons.

En analysant les programmes des trois élèves qui arrivent à une « utilisation partielle » de la communication par envoi de messages, nous pouvons identifier une erreur qui se répète. L'erreur de ces trois projets concerne la commande « envoyer message » et plus précisément, sa position sur le script du personnage « enfant ». Il est important de noter ici que les trois élèves choisissent d'ajouter la commande « envoyer message » sur le script du correct personnage.

Aucun de ces élèves, n'a eu de problèmes avec la commande « Quand message reçu ». La morphologie de cette dernière est telle que la syntaxe du logiciel oblige l'utilisateur à la poser au début des programmes de personnages. La position de la commande « envoyer message » sur le script du personnage « enfant » dépend d'une part, de la compréhension de la consigne du problème et d'autre part, de la compréhension de l'exécution séquentielle, qui caractérise l'environnement ScratchJr.

D'après nos observations de terrain, nous avons pu remarquer que la compréhension de l'exécution séquentielle était très importante, pour l'enseignement des autres notions présentes dans notre scénario pédagogique. Nous pouvons alors supposer que ces trois élèves sont arrivés à un tel résultat, soit parce qu'ils n'ont pas bien compris l'énoncé du problème, soit parce qu'ils n'ont pas bien compris le concept d'exécution séquentielle.

Les trois élèves restants n'utilisent pas les commandes de messages. Il s'agit de deux garçons et une fille. Nous pouvons supposer que ces élèves n'avaient pas compris la fonctionnalité des commandes des messages et c'est pourquoi ils ne sont pas arrivés à les utiliser dans leur projet. Selon une étude de Flannery et al. (2013), seulement les élèves les plus avancés de leur classe arrivent à manipuler de manière opérationnelle les messages.

Nos résultats sont similaires que ceux provenant de l'étude que nous avons réalisée auprès des élèves de cours préparatoires (Touloupaki et al., 2018). Les élèves de maternelle font la même erreur que ceux de cours préparatoires concernant le positionnement de la commande « envoyer un message ». D'autre part, la commande « quand message reçu » ne pose pas de problèmes aux élèves de ces deux niveaux. Un tel résultat est assez logique du moment où il est bien connu que, le niveau développemental des enfants est assez similaire entre ces deux groupes d'âges (5-6 ans et 6-7 ans). Le taux de réussite en termes de manipulation de la notion en question est aussi assez similaire. Pourtant, les élèves de cours préparatoires réussissent un peu mieux que les élèves de maternelle (6/12 pour l'école maternelle et 8/12 pour le cours préparatoire). Enfin, il convient de souligner l'importance du scénario pédagogique utilisé au cours de ce processus d'apprentissage. Les résultats de cette recherche sont dépendants du scénario pédagogique et des stratégies d'enseignement utilisées.

5 Discussion

Notre étude a eu un statut exploratoire et nous avons conscience qu'elle souffre de sérieuses limites. Tout d'abord, notre échantillon est de faible taille et nos observations ont duré seulement trois mois. Il n'est pas facile de mettre en place un scénario aussi élaboré dans des conditions réelles de classe. C'est pourquoi, nous avons eu besoin de nous adapter à la disponibilité du temps de l'enseignante de la classe et de réaliser onze séances au lieu de treize, ce qui était le programme initial.

D'autres recherches avec des échantillons plus larges seraient nécessaires, afin de pouvoir mieux étudier comment les jeunes enfants parviennent à se familiariser avec la notion de message.

Nous ne pouvons évidemment pas prétendre que les sujets considérés se sont approprié le concept de message, seulement qu'ils se sont montrés capables d'utiliser, dans les conditions d'un scénario donné, les commandes de messages de ScratchJr. Cela ouvre des possibilités dans la perspective de créer des programmes relativement sophistiqués pouvant gérer la communication entre entités différentes.

On a ainsi des perspectives permettant de diversifier à l'école primaire les possibilités de familiarisation pratique des jeunes avec l'informatique et de mettre en contact les élèves avec des notions qu'ils retrouveront sous une forme théorique plus tard dans leur scolarité. Il sera aussi nécessaire d'investiguer plus en détail les erreurs des élèves, afin de concevoir des moyens de les surmonter.

6 Références

1. Bers, M. U. : *Designing Digital Experiences for Positive Youth Development: From Playpen to Playground*. 1st edn. Oxford University Press, New York (2012).
2. Cobb, P., Confrey, J., DiSessa, A., Lehrer, R., & Shauble, L.: *Design Experiments in Educational Research*. *Educational Researcher*, 32(1), 9-13 (2003).
3. Fatourou, E., Zygouris, N., Loukopoulos, T., & Stamoulis, G.: *Teaching concurrent programming concepts using Scratch in primary school: Methodology and Evaluation*. *International journal of engineering pedagogy*, 8(4), 89-105 (2018).
4. Fessakis, G., Gouli, E., Mavroudi, E. : *Problem-solving by 5-6 years old kindergarten children in a computer programming environment : A case study*. *Computers & Education*, 63, 87-97 (2013).
5. Franklin, D., Conrad, P., Boe, B., Nilsen, K., Hill, C., Len, M., Dreschler, G., Aldana, G.: *Assessment of computer science learning in a scratch-based outreach program*. In *Proceedings of the 44th SIGCSE technical symposium on computer science education*, ACM, Denver (2013).
6. Flannery, L. P., Kazakoff, E. R., Bonta, P., Silverman, B., Bers, M. U., & Resnick, M.: *Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming*. In: *Proceedings of the 12th international conference on interaction design and children*, pp.1-10. ACM, New York, (2013).
7. Komis, V., Touloupaki, S, Baron, G.-L. : *Analyse cognitive et didactique du langage de programmation ScratchJr*. In : *Proceedings of the 6th conference Didapro-DidaSTIC*, pp. 109-121. Presses universitaires de Namur. Namur (2017).

8. Meerbaum-Salant, O., Armoni, M., Ben-Ari, M.,: Learning computer science concepts with Scratch. In: Proceedings of the 6th international workshop on computing education research, .pp. 69-76.ACM, New York,(2010)
9. .pp. 69-76.ACM, New York,(2010)
10. Nikolos, D., Komis, V.,: Synchronisation in Scratch : A case study with education science students. *Journal of computers in mathematics and science teaching* 34(2),223-241(2015).
11. Robert, F. (1985). L'utilisation de l'ordinateur dans l'enseignement primaire : L'exemple de la France. *Enfance*, 38(1), 19-30. <https://doi.org/10.3406/enfan.1985.2857>.
12. Touloupaki, S., Baron, G.-L., & Komis, V. : Un apprentissage de la programmation dès l'école primaire: le concept de message sur ScratchJr. In : Proceedings of the 7th conference Didapro-DidaSTIC, pp.303-323.Peter Lang AG, Bern Switzerland (2018).