

Tâches de programmation avec Scratch à l'école primaire : Observation et analyse du développement des compétences en mathématique

Gaëtan Temperman, Caroline Anthoons, Bruno de Lièvre, Joachim de Stercke

▶ **To cite this version:**

Gaëtan Temperman, Caroline Anthoons, Bruno de Lièvre, Joachim de Stercke. Tâches de programmation avec Scratch à l'école primaire : Observation et analyse du développement des compétences en mathématique. Frantice.net, 2014, TICE : scolarisation et évolution des pratiques, 9, http://www.frantice.net/document.php?id=1013. halshs-01092656

HAL Id: halshs-01092656
https://shs.hal.science/halshs-01092656

Submitted on 13 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Tâches de programmation avec Scratch à l'école primaire : Observation et analyse du développement des compétences en mathématique

Programmation tasks with Scratch : Observation and analysis of the progression in mathematics

Gaëtan Temperman, Caroline Anthoons, Bruno de Lièvre et Joachim de Stercke
Université de Mons, Département des sciences et de la technologie de l'éducation, Service de pédagogie générale et des médias éducatifs

Résumé

Cette contribution a pour objet d'une part de décrire un dispositif d'apprentissage en mathématique mobilisant le logiciel de programmation « Scratch » développé par le MIT et d'autre part, de rapporter des résultats quant aux effets de cet environnement sur la progression de la maîtrise des compétences des élèves à l'école primaire. Les études empiriques relatives à l'usage du Logo révèlent qu'il ne suffit pas de laisser l'élève livré à lui-même dans un micromonde pour observer des effets positifs sur le développement de ses compétences (Gurtner, 1991). Elles mettent en évidence que le rôle de l'enseignant est primordial à la fois en termes de conception de tâches didactiques et en termes d'encadrement de celles-ci. La scénarisation présentée dans cet article prend en considération cette recommandation. Elle s'appuie sur une dizaine de séquences d'apprentissage basées sur des situations problèmes guidées et pouvant être résolues à l'aide de la programmation de scripts dans Scratch. Si nous observons des gains d'apprentissage élevés dans le domaine des grandeurs, des nombres et de la structuration spatiale, l'impact du dispositif semble plus réduit pour les compétences relatives à la résolution de problèmes et pour le raisonnement logique. Un autre résultat intéressant qui ressort de notre analyse est l'effet du scénario en termes d'équité. Les résultats des apprenants sont en effet plus homogènes au terme de l'apprentissage.

Mots-clés : programmation, apprentissage du code, micromonde, constructionnisme, efficacité, équité, didactique de l'informatique et la mathématique

Summary

This contribution has two goals. On one hand, it aims to describe a learning environment in mathematics mobilizing the programming software "Scratch" developed by the MIT. On the other hand, it intends to bring back results as for the effects of this environment on the learners' skills at elementary school. Empirical studies on Logo emphasize letting the learner progress freely in a micro-world is not sufficient to observe positive effects on the development of his skills (Gurtner, 1991). These studies highlight the role of the teacher is essential in the design of didactic tasks as well as in their supervision. The pedagogical script presented in this article takes this recommendation in count. It is composed of ten learning scripts based on guided problem situations that can be resolved by programming in Scratch. If our analysis reveals learning gains in the fields of greatness, numbers and geometry, the impact of the pedagogical script seems lower for the problem solving skills and for the logical reasoning. Another interesting result is the effect of the script in terms of equity. The mastering level is indeed more homogenous at the end of learning.

Keywords: programmation, code learning, micro-world, constructionnism, efficacy, equity, didactics of informatics and mathematics

I. Introduction.

Qui ne connaît pas le langage Logo ? Développé par Papert (1981), ce langage de programmation pour les élèves a connu une diffusion importante dans le monde de l'éducation avec le succès des ordinateurs personnels dans les années 80 et 90. La programmation d'un script avec le langage Logo permettait de dessiner des figures géométriques et d'aborder par la même occasion les notions d'algorithme, de récursivité et de variables. Elle permet la création d'environnements autonomes appelés "micromondes" dans lesquels toutes les actions sont ordonnées et dirigées par des règles établies. Le langage utilisé a un rôle d'artefact dans la mesure où il médiatise l'interaction de l'apprenant avec son environnement. En amenant les élèves à développer ces micromondes, Papert (1981) formule ainsi l'hypothèse que ces tâches de programmation ont un impact positif sur les compétences cognitives des élèves. Les synthèses de Gurtner (1991) révèlent toutefois qu'il ne suffit pas de laisser l'élève livré à lui-même dans ce type d'environnement pour observer des effets sur le développement de ces compétences. Elles tendent plutôt à montrer que le rôle de l'enseignant est primordial à la fois en termes de conception de situations didactiques et d'encadrement de celles-ci. Bruillard (1997) rapporte des travaux de plusieurs auteurs traitant des dispositions à prendre au niveau pédagogique pour espérer un effet positif de l'utilisation de ces logiciels. Dans ce domaine, Carver (1986) constate que l'apprentissage d'aptitudes intellectuelles de haut niveau peut être effectif pour autant que les compétences les composant soient clairement spécifiées et qu'elles fassent l'objet d'un apprentissage direct. Dans ces conditions, le transfert peut être observé. De Corte & *al.* (1991) cités par Bruillard complètent ces résultats en montrant que si les apprenants obtiennent un niveau de maîtrise suffisant du Logo et des compétences en matière de programmation, le transfert à d'autres situations se produit sans entraînement supplémentaire. Pea, Kurland & Hawkins (1985) abondent dans le même sens en ajoutant que, dans la mesure où les enfants ont effectivement compris l'utilisation du langage de programmation, ils peuvent développer en parallèle des compétences en matière de résolutions de problèmes. Selon Swan (1989), le développement de compétences de résolutions de problèmes passe par une structuration pédagogique des tâches de programmation qui mobilisent différentes stratégies spécifiques chez les apprenants : la formation de sous-buts, le chaînage avant, les essais et erreurs systématiques et l'analogie. Les résultats obtenus indiquent un avantage obtenu du travail avec l'ordinateur par rapport à une situation de manipulation concrète traditionnelle, mais aussi l'apport d'un enseignement explicite par rapport à la découverte libre. Le langage Logo peut ainsi s'avérer efficace pour les élèves qui éprouvent des difficultés, auxquels le langage scolaire ne convient pas, il peut avoir des conséquences bénéfiques sur le comportement en classe (Béziat, 2012). Ces différents constats convergent vers l'idée que le rôle de l'enseignant est primordial dans les situations d'apprentissage utilisant le Logo. Afin de pouvoir espérer constater des effets sur le développement des compétences et le transfert de celles-ci à d'autres situations, il s'agit avant tout de ne pas laisser l'élève livré à lui-même devant l'ordinateur et par conséquent de structurer les tâches qu'il sera amené à réaliser.

Actuellement, un engouement important pour des démarches constructionnistes s'est fait jour comme en atteste le développement de nouveaux outils de programmation (Scratch, Lego Mindstorms EV3, Lego We Do, Sphero ...), la mise en place de nombreux projets mis en ligne[1] incitant à enseigner le code à l'école (Guillaud, 2014) ainsi que la volonté des politiques d'intégrer l'apprentissage de la programmation dans l'enseignement obligatoire. Cette idée ne fait toutefois pas l'unanimité. Ses détracteurs mettent en avant une perte de temps potentielle pour les apprentissages fondamentaux et l'absence d'indications précises dans les programmes d'études (Touret, 2014). D'un point de vue pédagogique, une solution pragmatique permettant de concilier ces deux points de vue nous semble être envisageable en nous appuyant sur les recommandations décrites ci-dessus et issues des expérimentations avec le langage Logo. Dans un premier temps, cette solution doit s'appuyer sur une scénarisation qui identifie dans un premier temps les compétences mathématiques du programme susceptibles d'être développées à l'aide de l'outil de programmation et les modalités d'évaluation de celles-ci. Dans un deuxième temps, cette structuration conduit à la conception de tâches spécifiques dans l'environnement de programmation, tâches permettant d'exercer et de mettre en œuvre lesdites

[1] http://code.org

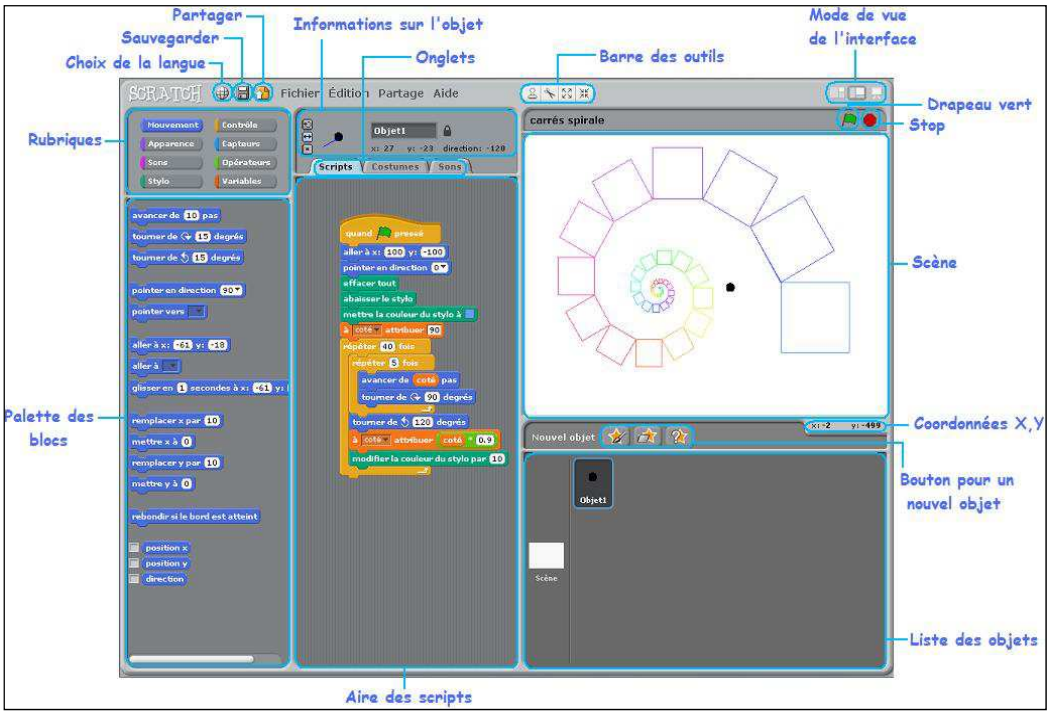
compétences. Dans le cadre de cette contribution, nous allons décrire un scénario pédagogique développé selon cette approche. Cet article sera également l'occasion de rapporter les principaux résultats en termes d'apprentissage à la suite de la mise en oeuvre de ce scénario dans une classe de 6e primaire de la Fédération Wallonie-Bruxelles (Belgique), obtenus à l'aide d'un plan expérimental de type prétest/posttest.

II. Le scénario pédagogique.

A. Description de l'environnement de programmation.

Notre choix s'est porté sur l'environnement Scratch dans la mesure où celui-ci est utilisé par une communauté de pratique importante qui permet de découvrir et de partager des projets de programmation. Développé par l'équipe de Michaël Resnick[2], Scratch s'appuie sur les bases de Logo, mais se révèle plus adapté aux possibilités graphiques des ordinateurs actuels (Resnick & *al.*, 2009). Il peut ainsi intégrer des médias différents et dispose d'une meilleure utilisabilité. Peppler & Kafai (2007) ont ainsi mené une étude quantitative sur les productions réalisées avec Scratch, et plus particulièrement sur la création de jeux vidéo. En observant les archives des serveurs des Computer Clubhouses deux ans après le lancement de Scratch, ils constatent qu'il est le logiciel de conception le plus populaire utilisé au cours de cette période et qu'il constitue la seconde activité au niveau de la popularité, la première étant la recherche sur le Web. Les auteurs avancent l'idée que la popularité de Scratch est due en grande partie à sa facilité d'utilisation et à sa flexibilité qui permet de varier le genre de jeux et de favoriser la créativité. Comme l'illustre la figure 1, l'interface ne demande plus à l'utilisateur de taper des lignes de code les unes à la suite des autres comme c'était le cas avec le langage LOGO.

Figure 1. Interface de Scratch



Basée sur la métaphore du jeu de construction Lego, la programmation avec Scratch consiste plutôt à assembler par "glisser-déplacer" et de manière logique des blocs associés à des fonctions spécifiques. Pour faciliter le travail de programmation, l'interface se compose de quatre parties distinctes : la

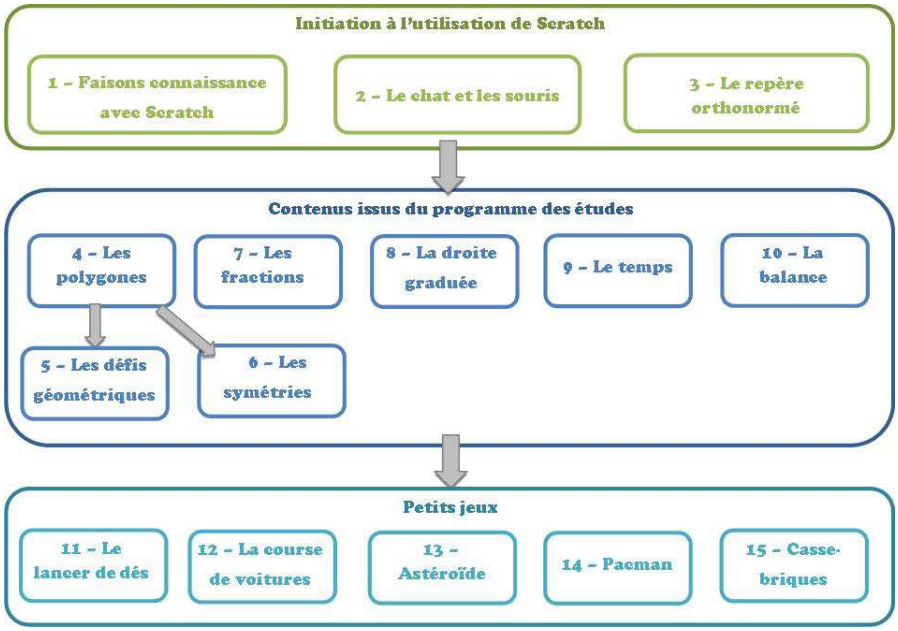
[2] http://scratch.mit.edu/

palette des blocs avec les différentes commandes de programmation disponibles (mouvement, apparence, sons, stylo, contrôle, etc.), la zone de travail pour assembler les blocs de programmation, la liste des objets programmables et enfin la scène pour visualiser le résultat. Parallèlement à cette facilité d'utilisation, le logiciel offre la possibilité d'expérimenter des fonctions complexes relatives à la pensée formelle comme les notions de séquence, d'itération, d'instructions conditionnelles, de variables, de synchronisation, d'opérations logiques booléennes. Brennan & Resnick (2013) se sont intéressés à la façon dont la conception d'activités d'apprentissage avec Scratch, en particulier la programmation de médias interactifs, peut soutenir le développement de la pensée algorithmique chez les jeunes. Ces auteurs observent quatre grands ensembles de pratiques chez les utilisateurs de Scratch : être progressifs et itératifs, pratiquer les tests et le débogage, adopter la réutilisation et le remixage, et enfin, utiliser l'analyse et la modularisation. Dans un contexte de formation universitaire, Baron & Voulgre (2013) ont quant à eux mis en évidence que les projets finaux sont généralement peu sophistiqués avec Scratch en termes de qualité de la programmation et recommandent de proposer une progression dans la maîtrise du logiciel envisagée sur une durée d'apprentissage significative.

B. Activités d'apprentissage.

Dans le cadre du développement du scénario pédagogique, nous avons tenu compte de ces observations réalisées par Brennan & Resnick (2012) et Baron & Voulgre (2013). Les activités d'apprentissage proposées dans le cadre de cette expérimentation s'appuient ainsi sur une gradation dans les niveaux de difficulté en référence à la taxonomie de Tirtiaux (1972) cité par Minder (2007). Cette taxonomie permet de moduler les tâches d'apprentissage en fonction du degré d'initiative laissé aux élèves. Dans cette progression, l'élève est guidé pour passer d'un mode de pensée convergente, basée sur l'application des connaissances et du raisonnement logique pour produire une réponse correcte (problème fermé) ; vers un mode de pensée divergente - fondée sur l'utilisation des connaissances, du raisonnement, et de la créativité afin de produire une réponse originale (problème ouvert). La taxonomie de Tirtiaux se structure autour de sept niveaux ordonnés (reproduire, reconnaître, ajuster, appliquer, achever d'initiative, imiter et inventer) allant du plus simple (initiative nulle) au plus complexe (initiative optimale).

Figure 2 : Enchaînement des activités pédagogiques



La figure 2 schématise l'articulation des différentes activités proposées. Notre dispositif comporte ainsi quinze activités spécifiques. Elles sont conçues pour être dispensées dans un ordre

prédéterminé. Premièrement parce que le niveau de difficulté, selon la taxonomie de Tirtiaux, est toujours croissant. La numérotation de scénarios met en évidence cette progression. Deuxièmement, parce que certaines notions spécifiques à Scratch sont traitées dans un premier scénario et utilisées par d'autres ensuite. Par ailleurs, les activités proposées ont été construites pour être significatives par rapport aux contenus présentés dans le programme des études de la Communauté française de Belgique. Ainsi, la première série de scénarios concerne directement les contenus scolaires du programme de mathématiques. La deuxième série propose des défis ludiques destinés à travailler la résolution de problèmes et le raisonnement logique. Dans le cadre de notre recherche, l'étude nécessite pour certaines variables de regrouper les scénarios pédagogiques d'après les différents domaines spécifiques présents dans le programme de mathématiques (Nombres et opérations, Grandeurs, Solides et figures, Situations problèmes). Même si dans une activité axée, par exemple, sur les solides et figures, l'élève exerce également ses compétences en grandeurs ou en nombres et opérations, nous rattachons ce scénario selon son objectif principal, c'est-à-dire le développement de compétences spécifiques à la géométrie. Le tableau 1 reprend cette catégorisation des différents scénarios pédagogiques en fonction de la compétence principale ciblée.

Tableau I : Catégorisation des scénarios pédagogiques selon les domaines mathématiques

Nombres et opérations	Grandeurs	Solides et figures	Situations problèmes
Les fractions	Le temps	Le repère orthonormé	Le lancer de dés
La droite graduée	La balance	Les polygones	La course de voitures
		Les symétries	Astéroïde
			Pacman
			Casse-briques

III. Méthodologie

A. Échantillon

Notre échantillon se compose de 19 élèves d'une classe de 6e primaire de la province de Hainaut en Belgique. Le choix de l'école a été réalisé en fonction de la disponibilité du matériel informatique et de l'accord des enseignants quant à la mise en place du dispositif. L'expérimentation a été menée sur une durée de 4 mois (de janvier à avril 2012). Au total, dix séances de deux heures ont pu ainsi être organisées et envisagées de manière collaborative par paires. Il est à noter qu'un élève, particulièrement performant, s'est retrouvé seul dans le cadre de l'expérience.

B. Plan expérimental

Dans l'optique d'évaluer l'effet du dispositif pédagogique mis en place sur le développement des compétences en mathématique des élèves, notre étude s'appuie sur un plan expérimental de type prétest / post-test. Ces tests ont permis d'obtenir une mesure du degré de maîtrise des élèves en mathématique et une mesure de leur niveau de raisonnement logique. Le prétest nous a également offert l'opportunité de constituer des paires hétérogènes composées d'un élève plus avancé et d'un élève moins avancé. Pour élaborer le test en mathématique, nous nous sommes basés sur les épreuves du Certificat d'Etudes de Base des années précédentes de manière à assurer une validité de contenu à notre outil. Nous avons sélectionné au sein des épreuves de 2010 et de 2011, les exercices les plus en adéquation avec les contenus que nous projetions de travailler en classe dans l'environnement Scratch. Notre questionnaire a été conçu pour garder un équilibre entre les quatre domaines du programme en mathématique : nombres et opérations, grandeurs, solides et figures, et situations problèmes. Concernant l'évaluation du raisonnement logique, nous avons porté notre choix sur le test de raisonnement différentiel de Meuris (1970). Il permet d'objectiver quatre dimensions complémentaires du raisonnement logique : hypothético-déductif, verbal, numérique et spatial. Ce dispositif d'évaluation ipsative nous donne la possibilité de calculer un gain relatif[3] pour chaque

[3] « Le gain relatif se calcule par la formule suivante (Score posttest - Score prétest) / (Score maximun - Score prétest) × 100. Il correspond au rapport de ce que l'élève a gagné à ce qu'il aurait pu gagner au maximum. Il est indépendant du

étudiant au terme de la formation. Pour compléter ces données relatives à la performance des élèves, nous avons proposé aux élèves un questionnaire de satisfaction au terme de l'apprentissage. Ils ont pu ainsi nous faire part de leur perception par rapport aux activités réalisées lors de l'expérimentation. Pour construire ce questionnaire, nous avons utilisé, comme modalité de réponse, une échelle de type "différenciateur" d'Osgood opposant un pôle négatif et un pôle positif sur laquelle les élèves positionnent une croix selon leur degré d'accord avec la proposition. Ce questionnaire comprend trois types d'items : une perception de leur progression dans l'apprentissage, une perception de l'utilité de la tâche pour apprendre et enfin une perception de l'attrait de la tâche. À partir de notre plan expérimental et des différentes variables dépendantes prises en considération, nous formulons les trois questions de recherche suivantes :

- **Q1** : *Comment les élèves évoluent-ils dans la maîtrise de compétences en mathématique ?*
- **Q2** : *Comment les élèves évoluent-ils dans la maîtrise de capacités en raisonnement logique ?*
- **Q3** : *Quelles sont les perceptions des élèves par rapport à leur expérience d'apprentissage ?*

IV. Analyse des résultats

A. Q1 : Comment les élèves évoluent-ils dans la maîtrise des compétences en mathématique ?

À la lecture du tableau 2, nous observons que la moyenne des gains relatifs des performances globales en mathématique est égale à 66,62 %. Les élèves ont donc atteint en moyenne les deux tiers de l'évolution possible. Cette progression est nettement supérieure à 30 %, valeur au-delà de laquelle on peut considérer qu'il y a effectivement eu apprentissage selon D'Hainaut (1975). Parallèlement à l'augmentation du niveau de maîtrise des élèves, il est intéressant d'observer que la dispersion (mesurée à l'aide du Coefficient de Variation) diminue de manière importante entre le prétest et le post-test. Il y a donc un plus grand équilibre de compétences entre les élèves au terme de l'apprentissage. Si nous croisons le prétest et le gain relatif, nous n'observons pas de lien significatif entre ces deux mesures ($r = .246$; $p = .310$). Cette analyse corrélationnelle nous semble intéressante à rapporter afin de déterminer à quel type d'élève profite le dispositif. Une corrélation significative et positive entre le gain relatif et le prétest nous indique que le dispositif a permis de faire progresser les plus avancés au départ. A contrario, une relation significative et négative nous informe que celui-ci a plutôt été bénéfique pour les moins avancés. Ici, l'absence de lien significatif tend à montrer que le dispositif profite à l'ensemble des élèves en mathématique, quel que soit leur niveau de départ.

Tableau II : Performances globales en mathématique

	Moy (%)	CV (%)
Score total au prétest	60.65	27.65
Score total au post-test	86.25	9.62
Gain relatif	66.62	23.77

Le tableau 3 permet d'affiner ce résultat et de distinguer les performances en fonction des différents domaines mathématiques pris en considération. La meilleure évolution (Moy = 82.27 %) et l'homogénéisation la plus importante des scores (CV = 9.80 %) s'observent dans les matières relatives aux solides et figures. Ce résultat s'accorde avec l'idée que le Logo se limite à l'apprentissage de la géométrie. On peut également avancer l'idée que le logiciel Scratch qui demande une organisation spatiale de l'écran d'animation sollicite indirectement la mobilisation de compétences géométriques. Si nous observons des gains élevés en numération (Moy = 61.18 %) et en grandeurs (Moy = 62.80 %), la progression en résolution de problèmes (Moy = 36.24 %) semble toutefois plus réduite comparativement aux autres domaines mathématiques. Cette différence constatée de progression entre les différentes matières peut s'expliquer par le type de transfert attendu par notre dispositif. En effet, pour les domaines liés aux nombres et opérations, aux

niveau de départ et comme, à niveau de départ égal, il est proportionnel à la performance. On peut considérer que le gain relatif est proportionnel à ce qu'il veut mesurer. », (p. 158-159).

grandeurs, et aux solides et figures, nous attendions un transfert horizontal des compétences. Il s'agissait de généraliser les connaissances à de nouveaux contextes relativement proches de ceux des situations d'apprentissage. Par contre, en ce qui concerne la résolution de situations problèmes, le transfert attendu était davantage de type vertical. Les activités d'apprentissage, proposées sous la forme de situations ludiques, sont très éloignées des situations de tests plutôt scolaires. Ces résultats corroborent ceux de Verschaffel, De Corte & Schrooten (1991) qui ont constaté qu'avec le Logo, les transferts observés sont principalement proches et spécifiques. Nous convergeons également vers les résultats de Kalelioglu & Gulbahar (2104) qui mettent en évidence que la programmation avec Scratch n'entraîne pas un développement plus important des compétences en résolution de problèmes.

Tableau III : Performances en fonction des domaines mathématiques

	Pré-test		Post-test		Gain relatif	
	Moy	**CV**	**Moy**	**CV**	**Moy**	**CV**
Nombres et opérations	63.40	32.01	83.40	14.26	61.18	60.75
Grandeurs	46.30	57.66	82.60	14.40	62.80	38.21
Solides & Figures	56.00	28.21	90.08	9.80	82.27	21.28
Résolution de problème	76.30	28.44	88.20	10.54	36.24	96.28

Dans le tableau 4, les corrélations entre le niveau de départ et le gain relatif pour chaque contenu mathématique tendent à montrer que le scénario profite davantage aux élèves moins avancés en grandeurs (r=-.428 ; p=.068) et en situations problèmes (r =-.671 ; p=.002). À l'inverse, il apparaît que ce sont les élèves qui ont une meilleure maîtrise au départ qui ont des gains les plus importants en ce qui concerne le contenu relatif aux solides et figures (r=.524; p=.017). L'absence de lien au niveau des nombres et opérations indique que le dispositif se révèle plutôt neutre pour le développement de cette compétence (r=.333 ; p= .164).

Tableau IV : Corrélations entre le niveau initial et les gains relatifs par domaines

	r	**P-value**
Nombres et opérations	**.333**	**.164**
Grandeurs	**- .428**	**.068**
Solides et figures	**.542**	**.017***
Situations problèmes	**-.671**	**.002***

B. Q2 : Comment les élèves évoluent-ils dans la maîtrise des capacités en raisonnement logique ?

Le tableau 5 permet de mettre en évidence que les élèves progressent en raisonnement logique, mais de manière significativement moins importante qu'au niveau de la maîtrise dans les compétences de base (F = 25.195 ; p = .000). La comparaison des coefficients de variation entre le prétest (CV = 27.80 %) et le post-test (CV = 17.52 %) montre également une évolution positive de l'équité au terme de l'apprentissage. Par ailleurs, nous remarquons que cette progression n'est pas liée au degré de maîtrise initial en logique (r = -. 012 ; p = .961).

Tableau V : Performances globales en raisonnement logique

	Moy	**CV**
Score total au prétest	**45.79**	**27.80**
Score total au post-test	**60.84**	**17.52**
Gain relatif	**27.74**	**36.91**

Si nous distinguons les différentes dimensions du raisonnement logique (tableau 6), seul le raisonnement hypothético-déductif est marqué par une évolution importante (Moy = 37.40 %), mais dans une mesure nettement moindre que ce que l'on a pu observer au niveau des compétences

mathématiques. Ce résultat peut s'expliquer par l'analogie des contenus entre les activités de programmation et le test de connaissances en mathématiques, et à l'inverse, par la nature transversale des items du test de raisonnement différentiel de Meuris (1970). Ces résultats vont dans le sens des idées de Swan (1989) selon lesquelles seul un enseignement explicite associé à la pratique de la programmation peut induire le développement de compétences dans la matière ciblée. Mais aussi de celles de Carver (1986) qui remarque que l'apprentissage d'aptitudes intellectuelles de haut niveau ne peut être effectif que si les compétences les composant sont clairement spécifiées et font l'objet d'un apprentissage direct. Dans ce cas, le transfert aurait pu être observé. Or, dans la mise en œuvre de notre dispositif expérimental, il n'y a eu aucun enseignement explicite concernant les compétences en raisonnement logique, l'objet de l'enseignement ciblé par nos scénarios pédagogiques étant exclusivement les compétences mathématiques issues du programme d'études. Lhermenier-Marinho (1994) cités par Bruillard (1997) ont également abouti à la conclusion que les activités de programmation en LOGO ne produisent pas un développement des capacités intellectuelles globales, mais plutôt une maîtrise d'habilités spécifiques aux mathématiques.

Tableau VI : Performances dans les différentes dimensions en raisonnement logique

	Pré-test		Post-test		Gain relatif	
	Moy	**CV**	**Moy**	**CV**	**Moy**	**CV**
Hypothético-déductif	62.76	30.52	77.48	13.88	37.40	44.41
Raisonnement verbal	36.00	36.44	49.28	28.40	21.49	60.07
Raisonnement numérique	31.80	43.01	51.16	29.16	28.16	57.59
Raisonnement spatial	52.64	37.99	65.48	22.66	26.97	57.76

Comme l'indique le tableau 7, la progression dans les différentes dimensions du raisonnement de logique n'est pas liée au niveau initial contrairement à ce que nous pouvions observer précédemment au niveau des différents contenus mathématiques. Dans l'environnement d'apprentissage proposé, les caractéristiques de départ ne modulent pas le développement du raisonnement logique. Nous pouvons donc mettre en avant la neutralité du dispositif en ce qui concerne la progression dans ces compétences transversales.

Tableau VII : Corrélations entre le niveau initial et les gains relatifs suivant les types de raisonnement

	r	**P-value**
Raisonnement hypothético-déductif	**-.265**	**.272**
Raisonnement verbal	**.307**	**.201**
Raisonnement numérique	**-.076**	**.757**
Raisonnement spatial	**-.052**	**.831**

C. Q3 : Quelles sont les perceptions des élèves par rapport à leur expérience d'apprentissage ?

Le tableau 8 permet de mettre en évidence que la satisfaction pour l'ensemble des activités proposées présente une distribution relativement homogène avec une moyenne de 52,64 % et une dispersion de 23,29 %. Les élèves ont donc de façon générale moyennement apprécié les activités proposées. Ils ont préféré les activités concernant les matières relatives aux solides et figures (Moy = 71.60 %) et aux grandeurs (Moy = 64.45 %). Les activités autour des nombres et opérations se placent légèrement au-dessus des 50 % de satisfaction avec une moyenne de 59.21 %. Par contre, il semble qu'ils n'aient pas autant apprécié les scénarios travaillant les compétences en résolution de problèmes, car ils obtiennent un score moyen de 33.90 %, ce qui semble cohérent avec la progression réduite que nous avons constatée dans cette matière. Ce résultat se révèle plutôt contre-intuitif. En effet, nous aurions pu penser que ces scénarios basés sur la construction de jeux interactifs auraient

été davantage appréciés par leurs aspects ludiques. Cet avis plus négatif peut être lié à la difficulté des activités dans la mesure où elles se situent hors de la zone proximale de développement des élèves. Si nous n'observons pas les résultats obtenus par Wilson, Hainey & Connely (2013) qui mettent en avant une perception positive de l'usage de Scratch par les élèves, nous pouvons expliquer cette différence par le degré de liberté plus important offert à ces apprenants lors de cette expérimentation.

Tableau VIII : Perceptions des élèves concernant l'attrait pour les tâches proposées

Attrait pour...	Moy	CV
... l'ensemble des scénarios	52.64	23.19
... les scénarios axés sur les nombres et opérations	59.21	31.85
... les scénarios axés sur les grandeurs	64.45	33.06
... les scénarios axés sur les solides et figures	71.60	25.11
... les scénarios axés sur les situations problèmes	33.90	43.18

En ce qui concerne l'utilité perçue, les résultats se révèlent également cohérents avec les progressions mesurées précédemment (tableau 9). Les élèves estiment que les situations avec Scratch liées aux problèmes sont peu utiles (Moy = 29.25 %). Il est intéressant d'observer que cet avis est plutôt hétérogène (CV = 54.46 %). L'aspect ludique de ces dernières activités n'a donc pas joué dans l'appréciation des élèves. Ceux-ci les ont probablement jugées trop difficiles. On peut penser que la difficulté de la tâche affecte la perception de contrôlabilité de celle-ci. D'autres activités plus simples visant le développement de ces compétences complexes, mais plus proches des contenus scolaires traditionnels auraient peut-être dû être proposées avant ces scénarios ciblant les compétences de résolution de situations problèmes. Pour les enseignants désirant utiliser Scratch dans leur pratique, nous pouvons leur recommander d'être attentifs au fait que leurs élèves aient atteint un niveau d'aisance suffisant avec le logiciel, avant de leur proposer ce genre de tâche plus complexe (Baron & Voulgre, 2013).

Tableau IX : Perceptions de l'utilité des scénarios pédagogiques proposés

Utilité...	Moy	CV
... globale pour l'ensemble des scénarios	50.97	25.09
... des scénarios axés sur les nombres et opérations	63.74	40.41
... des scénarios axés sur les grandeurs	67.66	30.38
... des scénarios axés sur les solides et figures	69.05	27.64
... des scénarios axés sur les situations problèmes	29.25	54.46

Quand on s'intéresse au sentiment de progression des apprenants (tableau 10), nous observons un avis plus positif (Moy = 71.17 %) et plus homogène (CV=28.52 %) par rapport à l'attrait et l'utilité. Curieusement, les élèves mettent en avant que leur progression est plus faible en solides et figures. Ce résultat pose véritablement question dans la mesure où il s'agit du contenu où leurs progrès effectifs sont les plus élevés dans le dispositif. L'absence dans le dispositif de tâches métacognitives spécifiques permettant aux élèves de prendre du recul par rapport à leur démarche d'apprentissage de manière explicite peut constituer une piste explicative à cette incohérence entre la progression perçue et la progression réelle.

Tableau X : Perception de la progression d'apprentissage

Sentiment de progression ...	Moy	CV
... globale	71.17	28.52
... en nombres et opérations	74.71	30.61
... en grandeurs	74.38	29.72
... en solides et figures	66.60	33.73
... en situations problèmes	69.82	36.17

V. Discussion et perspectives

Au niveau du développement des compétences globales en mathématiques, nous avons pu constater que mobiliser le logiciel Scratch dans un environnement pédagogique structuré conduit à un niveau d'apprentissage élevé. En l'absence de groupe témoin, nous sommes cependant dans l'incapacité d'affirmer que ces performances sont uniquement dues à l'utilisation de Scratch. En ce qui concerne les différentes matières spécifiques aux mathématiques, nous observons la présence d'un apprentissage effectif pour chacune d'elles, d'ampleur toutefois variable. Les solides et figures présentent le gain le plus important, ce qui nous amène à formuler l'hypothèse que les compétences s'y rattachant ont été celles qui ont été le plus travaillées par notre dispositif. Quel que soit l'objet du projet, le logiciel Scratch demande en effet une organisation spatiale de l'écran qui sollicite indirectement la mobilisation de compétences géométriques. Sur un autre plan, nous constatons que les progressions relatives à la résolution de problèmes sont plus réduites, alors que ces compétences tiennent une place centrale dans les objectifs émis par Papert (1981). Nos résultats convergent donc avec les résultats de Valcke (1991) observés dans un contexte de programmation en Logo ainsi et ceux de Kalelioglu & Gulbahar (2014) avec Scratch qui mettent en évidence que les tâches de programmation ont finalement peu d'impact sur la résolution de problèmes. Cette différence de progression dans les divers contenus que nous avons observée peut selon nous s'expliquer par le type de transfert attendu par notre dispositif expérimental. Pour obtenir des effets plus tangibles des micromondes sur les compétences en résolution de problèmes, il importe probablement comme le suggère Valcke (1991) d'intégrer ces activités de programmation dans le curriculum de mathématique. Cette approche pourrait alors s'appuyer sur un développement spiralaire de ces compétences. En ce qui concerne le raisonnement logique, nous avons pu observer un apprentissage effectif pour les capacités hypothético-déductives et spatiales. La période d'expérimentation relativement courte nous paraît toutefois insuffisante pour attendre un apprentissage plus important dans les deux autres dimensions du raisonnement logique à savoir le raisonnement verbal et le raisonnement numérique. Nous retrouvons ici aussi une attente envers un transfert vertical, sur des contenus très éloignés de ceux finalement présentés dans les situations d'apprentissage.

Un autre résultat qui nous semble important à discuter est que les progressions positives observées s'accompagnent d'une meilleure équité entre les élèves au terme de l'apprentissage. Ce résultat est important à prendre en considération dans la mesure où la gestion de l'hétérogénéité constitue souvent une démarche difficile à mettre en œuvre pour les enseignants. Nous pouvons avancer comme hypothèse explicative que ce résultat est lié à la constitution hétérogène de nos paires collaboratives. Les analyses croisées entre le prétest et les progrès effectués dans les différents contenus se révèlent complémentaires à cet impact sur l'équité. Elles nous donnent en effet la possibilité d'évaluer d'une certaine manière l'interaction entre le dispositif et le niveau initial. Les corrélations significatives et négatives au niveau de la résolution de problèmes et des grandeurs indiquent que les activités mobilisant Scratch dans un contexte d'hétérogénéité bénéficient aux plus faibles. Des tâches de programmation proposées au mode collaboratif peuvent donc jouer un rôle de remédiation en grandeurs et en résolution de problèmes. Ce résultat converge vers les propositions de Béziat (2012) qui suggèrent l'impact positif du codage pour les élèves en difficulté d'apprentissage. À l'inverse, le lien positif et significatif mesuré au niveau des solides et figures indique que ce sont les plus avancés qui progressent davantage dans ce domaine. Cette observation laisse plutôt à penser que le dispositif pédagogique induit un « effet tuteur ». Cette formulation d'explications par les élèves plus avancés aux moins avancés entraîne une maîtrise plus importante des compétences en géométrie pour ceux-ci.

En termes de perspectives, nous avons le projet de poursuivre l'intégration d'autres outils de programmation dans des contextes d'apprentissage. Pour aboutir à une meilleure compréhension du transfert de compétences en mathématique et en raisonnement logique, il nous paraît toutefois important de prendre en considération les variables liées au processus d'apprentissage. Nous pensons en particulier à l'observation et à l'analyse des interactions entre les élèves lors de ces activités de programmation.

Références

Béziat, J. (2012). Les TIC à l'école primaire en France : informatique et programmation, *Revue de l'EPI*. Récupéré du site de la revue : http ://www.epi.asso.fr/revue/articles/a1311d.htm

Baron, G. et Voulgre, E. (2013). *Initier la programmation des étudiants de master de sciences de l'éducation ?* Un compte rendu d'expérience. Sciences et technologies de l'information et de la communication (STIC) en milieu éducatif, Clermont-Ferrand, France.

Brennan, K. & Resnick, M. (2013). Imagining, creating, playing, sharing, reflecting: How online community supports young people as designers of interactive media. In N. Lavigne & C. Mouza (Eds.), *Emerging Technologies for the Classroom : A Learning Sciences Perspective*. New-York : Springer

Bruillard, E. (1997). *Les machines à enseigner*. Paris : Editions Hermès.

D'Hainaut, L. (1975). *Concepts et méthodes de la statistique*. Bruxelles : Labor.

Guillaud, H (2014). Enseigner le code à l'école ? Vraiment ? *InternetActu*. Récupéré le 23 avril 2014 sur : http ://www.internetactu.net/2014/04/23/enseigner-le-code-a-lecole-vraiment/

Gurtner, J.-L. et Retschitzki, J. (1991). *LOGO et apprentissages*. Neuchâtel : Delachaux et Niestlé.

Kalelioglu, F. & Gulbahar, Y. (2014). The effets of teaching programming via Scratch on Problem Solving Skills : a discussion from learners 'perspective. *Informatics in Education*, *13* (1), 33-50.

Carver, S. (1986). Transfer of Logo debugging skill : Analysis, Instruction and assessment. Ph.D dissertation, Carnegie-Mellon University.

Meuris, G. (1970). *Test de raisonnement différentiel : manuel*. Bruxelles : Editest.

Minder, M. (2007). *Didactique fonctionnelle*. Bruxelles : De Boeck.

Papert, S. (1981). *Jaillissement de l'esprit, ordinateurs et apprentissage*. Paris : Flammarion.

Pea, R. D., Kurland, D. M., & Hawkins, J. (1985). Logo and the development of thinking skills. In M. Chen & W. Paisley (Eds.), *Children and microcomputers: Formative studies* (p. 193–212). Beverly Hills, CA: Sage.

Peppler, K. A. & Kafai, Y. B. (2007). *What video game making can teach us about learning and literacy : Alternative pathways into participatory cultures* ? Communication présentée au congrès the Digital International Games Research Association meeting in Tokyo, Japan.

Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52 (11), 60-67.

Swan, K. (1989). Logo programming and the teaching and learning of problem solving. *Journal of Artificial Intelligence in Education*, *1* (1), 73-92.

Tourret, L. (2014). Plus de programmation à l'école, d'accord, mais on la cale où dans le programme ? *Slate.fr*. Récupéré le 22 janvier 2014 : www.slate.fr/story/82571/programmation-programme-ecole

Valcke, M. (1991). Micro-mondes et mathématiques dans les degrés 3 à 6. Dans Gurtner, J-L. et Retschitzki, J. (dir.), *LOGO et apprentissages* (pp. 114-124). Dijon-Quetigny : Editions Delachaux & Niestlé.

Verschaffel, L., De Corte, E. et Schrooten, H. (1991). Transfert des stratégies cognitives par un système didactique basé sur LOGO. Dans J-L. Gurtner et J. Retschitzki (dir.), *LOGO et apprentissages* (p. 29-37). Dijon-Quetigny : Editions Delachaux & Niestlé.

Wilson, A., Hainey, T., Connolly, T.M. (2013). Using Scratch with primary school children: an evaluation of games constructed to gauge understanding of programming concepts. *International Journal of Game-Based Learning*, *3* (1), 93–109.