

Master of Arts en enseignement pour le degré secondaire I

Synthèse du Mémoire de Master

Réalisation et mise en pratique d'un dispositif permettant le développement de compétences dans le domaine de la pensée computationnelle au secondaire I

Auteur	David Morisod
Directeur	Prof. Eric Sanchez
Date	Septembre 2019

Introduction

Le développement rapide des technologies numériques conduit à une part toujours plus croissante du monde du digital dans les activités professionnelles et privées. De ce fait, un fossé tend à se creuser entre le degré de complexité des produits apparaissant sur le marché et les connaissances des utilisateurs. Ceci conduit la formation des futurs utilisateurs et créateurs à une évolution nécessaire. Ainsi, l'école obligatoire commence à modifier ses plans d'étude afin d'intégrer de nouveaux domaines du monde de l'informatique.

Ainsi, le sujet de ce travail porte sur l'enseignement de l'informatique et plus précisé-

ment de la pensée computationnelle. Cette pensée a été décrite pour la première fois à la fin des années 1960 puis a fortement été développé de manière théorique au début des années 2000. Elle se définit comme une pensée qui s'intéresse à la résolution de problèmes ou à la conception de systèmes en s'appuyant sur des compétences cognitives reprenant des concepts fondamentaux de l'informatique théorique. Les concepts retenus dans ce travail sont les suivants :

- Abstraction
- Algorithme
- Contraintes des systèmes (efficacité, performance, ...)
- Contrôle de flux
- Décomposition, classification et priorisation
- Généralisation et reconnaissance de modèles
- Itération
- Logique conditionnelle
- Systèmes et représentations symboliques
- Traitement systématique de l'information

Méthode

Dans cette recherche, un cours de programmation et robotique a été créé. Afin de mieux renforcer le développement de la pensée computationnelle chez les élèves, chacune des leçons a été élaborée de manière à introduire et entraîner les différentes composantes (citées ci-dessus) de la pensée computationnelle. Ce cours est structuré autour de huit leçons, réparties sur une vingtaine de périodes d'enseignement. Il comprend des activités débranchées ainsi que des ateliers de programmation de robots Thymio. Deux types de programmation y sont abordés : la programmation visuelle, au moyen de blocs représentatifs des différentes parties du robot et la programmation semi-textuelle dans laquelle la symbolique des blocs est conservée, mais les icônes sont remplacées par du texte définissant les différentes fonctions utilisables.

Le cours a été dispensé auprès de trois classes, réparties dans trois établissements différents (le CO du Belluard à Fribourg, le Collège du Reposieux de Monthey et le CO de Collombey). Les classes de Fribourg et de Monthey étaient composées d'élèves de 11H et celle de Collombey, d'élèves de 9H. De plus, les trois enseignants ayant accepté de participer à cette recherche ne possédaient pas de connaissances préalables approfondies

en programmation et robotique. De ce fait, le support de cours proposé est demeuré flexible du point de vue des adaptations possibles à apporter. Il était en effet important que chaque enseignant puisse s'approprier personnellement la matière tout en respectant les objectifs d'apprentissage de chaque leçon. Toutefois, afin de correspondre au mieux à la réalité de l'enseignement dispensé, un échange régulier a été mis en place avec chacun des enseignants. Ceci a conduit à des adaptations spécifiques pour chacune des classes en fonction des modalités de suivi du cours (nombre de périodes hebdomadaire, effectifs des classes, ...) ainsi que de l'âge et du niveau des élèves. Par ailleurs, trois classes contrôle ont également été sélectionnées. Il s'agissait de classes parallèles aux classes test (de ce fait équivalentes en termes d'âge et de niveau scolaire), dont l'unique différence au niveau de la pensée computationnelle est de ne pas avoir reçu d'enseignement spécifique dans ce domaine.

L'évaluation du dispositif s'est opérée grâce à deux test transmis aux élèves de chacune des classe test et contrôle. En premier lieu, un pré-test a servi à évaluer leur niveau initial dans le domaine de la pensée computationnelle. Ensuite, à la fin du cours, les élèves ont passé un post-test similaire au pré-test. Ces tests sont composés d'une série de situations-problème mobilisant, pour leur résolution, l'utilisation des différentes composantes de la pensée computationnelle. Ainsi, de par leur structure, ces tests permettent d'obtenir à la fois un score général pour l'entier des questions et également un score spécifique pour chacune des composantes évaluées. En effet, chacune des questions posées a été évaluée selon les différentes composantes qu'elle travaille. Les points ainsi obtenus peuvent donc être attribués précisément à chacune des composantes abordées dans la question. L'analyse des résultats s'est opérée sur la base non pas des scores bruts obtenus par les élèves, mais des variations entre le pré-test et le post-test. Ces variations (données en pourcentage) ont été calculées selon la formule suivante :

$$variation = (score_{post-test} - score_{pré-test}) / score_{pré-test} \cdot 100$$

De ce fait, le score de l'élève par rapport à la moyenne de sa classe ne revêt que peu d'importance. En effet, c'est la variation de ses résultats entre le pré-test et le post-test qui est étudiée et analysée dans ce travail.

Ainsi, l'objectif de cette recherche est de déterminer si le cours de programmation et robotique créé permet une amélioration des compétences des élèves dans le domaine de la pensée computationnelle. De manière plus opérationnelle, les hypothèses posées postulent que l'augmentation des résultats obtenus par les élèves au post-test est due au cours de programmation et robotique suivi.

Résultats

L'analyse préliminaire des résultats a permis de conclure à une répartition des résultats des élèves selon une loi normale de distribution. Ce constat est un prérequis aux calculs statistique effectués par la suite. Concernant les résultats obtenus, deux analyses ont été opérées. Le première s'est focalisé sur les résultats généraux des élèves aux tests. Ainsi, la variation des résultats a été comparée entre les classes test et les classes contrôle. À titre d'exemple, la figure suivante illustre la distribution des variations des résultats entre le pré-test et le post-test pour les élèves des classes test et contrôle de Fribourg.

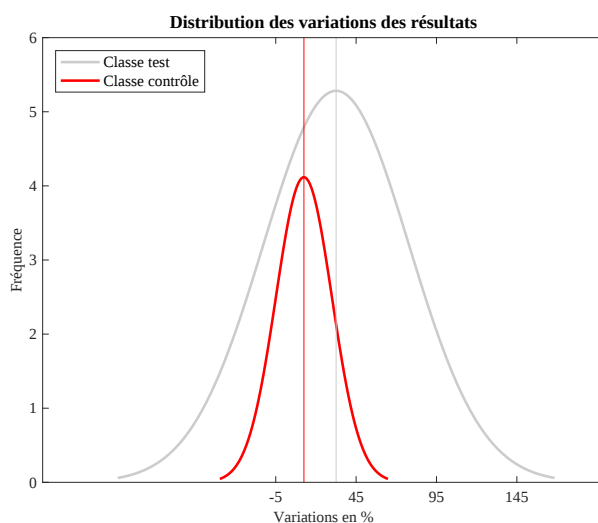


FIGURE 1 – Distribution des variations des résultats entre le pré-test et le post-test pour les classes de Fribourg

Cette figure traduit une augmentation des résultats plus marquée pour les élèves de la classe test que pour ceux de la classe contrôle (moyenne supérieure). Au niveau inférentiel, ces résultats ne sont pas significatifs pour la classe test de Monthey (qui n'a pas suivi les deux dernières leçons du cours) ainsi que pour celle de Collombey. Concernant la classe test de Fribourg, les résultats sont significatifs à 90%.

La seconde analyse a traité les résultats partiels obtenus pour chacune des composantes de la pensée computationnelle. Au niveau qualitatif, il en ressort le tableau suivant.

	Composantes									
Établissement	1	2	3	4	5	6	7	8	9	0
Fribourg	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗
Collombey	✗	✓	✗	✗	✓	✗	✓	✓	✗	✓

TABLEAU 1 – Résumé qualitatif des variations selon les composantes de la pensée computationnelle

Dans ce tableau, les ✓ indiquent une meilleure amélioration des résultats pour la classe test que pour la classe contrôle. Il permet de constater que les composantes *d'algorithme* (n° 2), de *décomposition* (n° 5), *d'itération* (n° 7) et de *logique conditionnelle* (n° 8) connaissent une amélioration dans les deux classes test ayant suivi l'intégralité du cours. En reprenant l'analyse des activités proposées dans les leçons, il est intéressant de constater que ces quatre composantes figurent parmi les cinq les plus travaillées et les plus développées dans le cours. Toutefois, ces résultats demeurent uniquement descriptifs car l'analyse inférentielle ne permet pas de conclure à une significativité statistique.

Conclusion

Ce travail constitue une approche différente de l'enseignement de l'informatique par l'apprentissage de la pensée computationnelle. Il a démontré un impact du dispositif proposé, sans toutefois parvenir à une significativité statistique. Par ailleurs, divers points ont pu être relevés afin d'améliorer le cours créé. Ainsi, une adaptation du support de cours en fonction de l'âge et du type de classe serait pertinente. De plus, l'ajout d'exercices de réinvestissement et de renforcement offrirait aux élèves un meilleur ancrage de leurs apprentissages.

Au regard des avancées techniques et technologiques, une évolution de l'enseignement de l'informatique à l'école obligatoire est inévitable. En ce sens, ce travail peut fournir une approche pour le domaine spécifique qu'est la pensée computationnelle.

Bibliographie sélective

- Berry, M. (2015). *QuickStart Primary Handbook*. Swindon : BCS.
- Calmet, C., Hirtzig, M. & Wilgenbus, D. (2016). *1, 2, 3 ... Codez!* Editions le Pommier.
- CIIP. (2010). *Plan d'Études Romand : MITIC*. Conférence intercantonale de l'instruction publique de la Suisse romande et du Tessin.
- Gouws, L., Bradshaw, K. & Wentworth, P. (2013). First Year Student Performance in a Test for Computational Thinking. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference* (p. 271-277). SAICSIT '13. East London, South Africa : ACM. doi :10.1145/2513456.2513484
- Hubwieser, P. & Mühling, A. (2015). Investigating the psychometric structure of Bebras contest : towards mesuring computational thinking skills. In *2015 International Conference on Learning and Teaching in Computing and Engineering* (p. 62-69). IEEE.
- Kradolfer, S., Dubois, S., Riedo, F., Mondada, F. & Fassa, F. (2014). A sociological contribution to understanding the use of robots in schools : the thymio robot. In *International Conference on Social Robotics* (p. 217-228). Springer.
- Lee, I., Martin, F., Denner, J., Coulter, B., Allan, W., Erickson, J., ... Werner, L. (2011). Computational thinking for youth in practice. *Acm Inroads*, 2(1), 32-37.
- Lockwood, J. & Mooney, A. (2018). Developing a Computational Thinking Test using Bebras problems. In *TACKLE : The 1st Systems of Assessments for Computational Thinking Learning workshop at EC-TEL 2018, At Leeds, United Kingdom*.
- Mubin, O., Stevens, C. J., Shahid, S., Al Mahmud, A. & Dong, J.-J. (2013). A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1(209-0015), 13.
- SEnOF. (2013). *MITIC - Informatique Planification annuelle fribourgeoise*. Service de l'enseignement obligatoire de langue française.
- Wing, J. M. (2006). Computational Thinking. *Commun. ACM*, 49(3), 33-35. doi :10.1145/1118178.1118215
- Yadav, A., Zhou, N., Mayfield, C., Hambruch, S. & Korb, J. T. (2011). Introducing computational thinking in education courses. In *Proceedings of the 42nd ACM technical symposium on Computer science education* (p. 465-470). ACM.