

Les micromondes de programmation : état de l'art

Fahima Djelil¹

¹ Université de Haute-Alsace, UHA 4.0, LMIA, 68200 Mulhouse, France
fahima.djelil@uha.fr

Résumé. Cet article contribue à la définition des micromondes de programmation en s'intéressant à la genèse des micromondes, en général, et à leur rapport avec la programmation, en particulier. Nous définissons les micromondes de programmation en les présentant comme des systèmes de représentation transitionnels. Ils présentent la propriété de représenter des concepts de programmation formels au moyen d'objets dits « transitionnels », car ils permettent à l'apprenant d'accéder à ces concepts de programmation en manipulant des objets concrets avec lesquels ils sont sémantiquement liés.

Mots-clés. Micromondes de programmation, Didactique de l'informatique.

Abstract. This article contributes to the definition of programming microworlds while focusing on the microworlds' genesis, in general, and on their relation to programming, in particular. We define programming microworlds as transitional representation systems. They have the property to represent formal programming concepts by means of objects called "transitional", since they provide the learner with the ability to understand the programming concepts by manipulating concrete objects to which they are semantically linked.

Keywords. Programming microworlds, Computer Science Education.

1 Introduction

L'apprentissage et l'enseignement de la programmation a suscité le questionnement de plusieurs auteurs [1] [2] [3]. De même, l'enseignement de la programmation est considéré comme l'un des défis majeurs de la didactique de l'informatique [4]. En particulier, l'enseignement des fondamentaux de la programmation au niveau universitaire a fait l'objet de nombreuses discussions impliquant des chercheurs, des enseignants et des didacticiens [5] [6] [7] [8]. À titre indicatif, l'interrogation de la base de données ACM Digital Library sur le terme CS1¹ renvoie 983 résultats et 97,364 résultats sur l'expression « introductory programming » (introduction à la programmation) [recherche réalisée le 23 Mai 2016]. Aujourd'hui, justifiées par l'émergence d'une culture numérique, les questions relatives à l'enseignement de l'informatique qui passe par l'initiation à la programmation, s'étendent à tous les niveaux scolaires et sortent du contexte universitaire.

¹ CS1 (Computer Science 1) désigne la discipline de l'informatique au premier cycle universitaire dans le curriculum anglo-saxon.

L'une des ambitions de certains chercheurs est de rendre la programmation à la fois accessible et attrayante quel que soit le niveau scolaire. Il s'agit de contribuer au développement d'une approche visant, d'une part, à rendre les concepts de programmation concrets et donc faciles à comprendre, et d'autre part, à rendre la programmation attrayante, et donc augmenter l'intérêt de l'apprenant pour cette discipline [9] [10] [11] [12] [13]. C'est dans cette perspective que plusieurs micromondes fondés sur la programmation tangible et la programmation visuelle ont été conçus [14] [15] [16] [17] [18]. Il s'agit d'utiliser des objets physiques ou graphiques pour représenter les concepts de programmation. Les micromondes se caractérisent en plus de leur capacité à présenter les concepts abstraits de façon concrète, d'une dimension de jeu offrant des contextes significatifs et attrayants conduisant à la construction active des apprentissages.

Ce papier a pour objectif de présenter le concept des micromondes à travers leur genèse et leur lien avec l'apprentissage de la programmation. Il est organisé comme suit. La section 2 donne la genèse des micromondes. La section 3 décrit leur rapport à la programmation. La section 4 introduit le concept de systèmes de représentation transitionnel, une formulation que nous proposons pour décrire les micromondes de programmation. Enfin, la section 5 conclut sur notre contribution dans la définition de ce concept.

2 Définitions et genèse des micromondes

Plusieurs auteurs ont contribué à la définition de la notion de micromonde [19] [20] [21] [10] [22] [23]. En particulier, leur genèse est décrite dans le livre : « *Artificial Intelligence and Education : Learning environments and tutoring systems* » [24], où il est rapporté que l'origine du mot micromonde remonte à un programme de reconnaissance de langue anglaise qui instruit un robot appelé SHRDLU, opérant dans un domaine miniature en manipulant des blocs physiques sur une table [25]. Ce mot a été ensuite introduit dans un rapport interne du MIT : « *The 72' progress report* » [26], dans lequel le terme désignait une subdivision de domaines de résolution de problèmes en fragments plus petits, ainsi que les schèmes qui peuvent se former lors de l'interaction avec ces fragments. Le terme a par la suite été popularisé après l'apparition de l'environnement LOGO et avec l'édition du livre : « *Mindstorms : Children, computers, and powerful ideas* » [10], pour désigner des domaines d'apprentissage devant être conçus comme génétiques, en faisant référence à la genèse et à l'évolution de la connaissance, incluant ce qu'un individu peut apprendre et comment il peut l'apprendre. Ceci est fondé sur le concept d'« *épistémologie génétique* » [27], qui implique essentiellement l'idée que l'individu acquiert de la connaissance en la construisant lui-même [10].

Alors que l'intention constructiviste des micromondes a été soulignée [10], ces derniers sont également décrits comme des mondes virtuels pour l'action créative: « *une idée essentielle est de créer un environnement avec lequel d'autres personnes pourront exercer leur créativité* » [21]. Cet environnement est constitué d'objets dits « *transitionnels* » [21] [10], car ils aident à développer de façon incrémentale des compétences impliquant la capacité à manipuler des objets formels [21]. Les objets

constituant un micromonde ont des propriétés communes à la fois avec les objets formels de la science et les objets plus concrets de l'expérience sensible ou du monde réel [21] [28]. Ils sont décrits comme des objets qui, d'une certaine façon, sont semblables à ceux avec lesquels on travaille dans le monde réel et, d'une autre façon, sont semblables à des objets abstraits : « *ce sont des objets qui aident à manipuler les objets abstraits* » [28]. De même cette capacité de créer des objets transitionnels, est une manière de combler l'écart entre l'apprentissage intuitif et l'apprentissage formel [28].

Un micromonde est également décrit comme un système constitué d'objets, de relations entre objets et d'opérations transformant ces objets et relations, pouvant servir à créer d'autres objets [23]. Ces objets sont visualisés graphiquement et constituent un modèle de concepts ou des concepts proposés à l'apprenant. Un micromonde donne forme à la structure d'un concept. La tâche de l'apprenant est d'intégrer mentalement cette structure afin de se l'approprier. Il constitue de ce fait, le cœur d'un système axiomatique intuitif, dans lequel les apprenants peuvent définir de nouveaux objets et opérations et étudier interactivement leurs propriétés [23].

Dans le même ordre d'idées, l'expression « micromonde transitionnel » a été proposée pour désigner les premiers micromondes qui ont été conçus, dont notamment LOGO [19]. Un micromonde est décrit comme un monde artificiel dans lequel on agit sur des objets, dont le comportement respecte certaines contraintes de fidélité et de cohérence [19]. L'idée essentielle sous-jacente à cette définition, est la fidélité et la consistance d'un certain modèle intégré et exprimé dans le micromonde. Un micromonde tente d'établir un lien sémantique fort entre le formel (ou l'abstrait) et le réel (simulé ou de référence), en garantissant la conservation du sens grâce aux connaissances de l'apprenant sur le fonctionnement des objets réels [19]. Un schéma général des différents éléments constitutifs des micromondes transitionnels a été proposé [19]. Dans ce schéma, le monde réel ou de référence est relié au monde formel ou abstrait par un processus de modélisation ou d'exemplification (Fig.1). Les objets visibles à l'interface se comportent d'une manière cohérente avec le modèle formel sous-jacent et leur aspect externe, c'est-à-dire à l'interface, rappelle les objets du monde réel. Ce schéma postule l'existence de modèles mentaux sur lesquels les micromondes sont fondés. Un modèle mental est vu comme une représentation dont les objets symboliques se comportent d'une manière similaire aux objets dans les situations où ils sont représentés [19]. Ces modèles mentaux évoluent par expérience dans les environnements proposés [19] [20] [22].

Une définition plus récente d'un micromonde le décrit comme un environnement numérique interactif qui permet à l'utilisateur d'interagir avec des modèles de situations et de phénomènes [29]. Cet environnement génère un feedback dynamique basé sur un ensemble de règles sous-jacentes, modèles et opérations comme réponses aux manipulations de l'utilisateur des objets et paramètres constituant cet environnement [29]. Cette définition est en lien avec celle qui décrit un micromonde comme un environnement d'apprentissage interactif, qui est un modèle conceptuel de certains aspects du monde réel [20]. Il s'agit d'un environnement qui est souvent idéalisé et simplifié, dans lequel les apprenants explorent ou manipulent la logique, les règles ou les relations du concept modélisé, tel qu'il est défini par le concepteur. En ce sens, un micromonde est vu comme un moyen de simplifier et de concevoir des

modèles du monde réel, permettant aux apprenants de manipuler et d'observer individuellement certaines contraintes et variables [20].

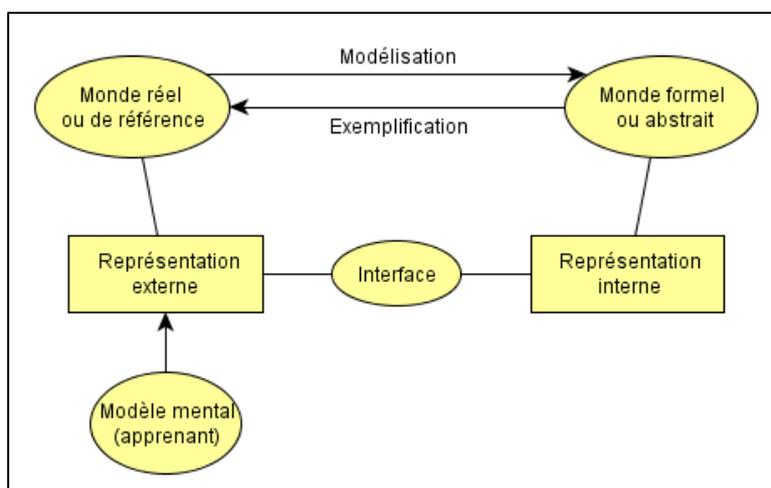


Fig.1. Schéma général d'un micromonde transitionnel [19].

Les micromondes ont toujours présenté un réel potentiel sur l'apprentissage de la programmation [10] [13]. En particulier, la programmation tangible et la programmation visuelle sont nées avec les micromondes. Dans ce qui suit nous nous intéressons au rapport des micromondes à la programmation visuelle et tangible.

3 Rapport des micromondes à la programmation

La programmation tangible et/ou visuelle fait partie intégrante des premiers micromondes qui ont été conçus, dont notamment LOGO. La programmation dans l'environnement LOGO, est introduite à travers une métaphore qui est d'instruire une tortue robotique ou graphique en lui apprenant de nouveaux mots [10]. Les utilisateurs du système LOGO commencent leur expérience de programmation en programmant la tortue de sorte à ce qu'elle réponde à de nouvelles commandes. Il a été observé que la programmation en LOGO a permis aux élèves de comprendre de façon précise un certain nombre de concepts mathématiques et physiques, car ils sont amenés à comprendre leurs propres processus de pensée en verbalisant leurs expériences [10]. Au-delà des connaissances apprises par la programmation, les apprenants apprennent également la programmation. Plus encore, l'utilisation de LOGO ne peut se passer de l'apprentissage de la programmation. Il a été souligné qu'une connaissance minimale de la programmation est indispensable pour savoir utiliser cet outil, comprendre son but, interpréter les erreurs et savoir les corriger [30]. Un compromis est à trouver entre l'apprentissage de l'outil et les services qu'il peut rendre. Mais ce qui était un inconvénient est vite devenu un avantage. En effet, l'apprentissage par la programmation peut s'accompagner ou engendrer de nouveaux savoirs et savoirs-faire en programmation [19]. L'apprentissage des aspects

techniques de la programmation était alors devenu l'objet principal du projet LOGO [30]. Ce qui était initialement conçu comme outil d'apprentissage par la programmation est devenu un outil d'apprentissage de la programmation. C'est aussi le cas de l'environnement Alice qui a été initialement conçu dans le but de simplifier la programmation 3D aux non-programmeurs [31]. Il a été ensuite exploité comme support d'apprentissage de la programmation [14]

En dehors de LOGO, d'autres environnements utilisant le langage LOGO, ou se basant sur le même principe que LOGO, ont été présentés comme de bons supports à l'apprentissage de la programmation [14] [16] [32]. Leur objectif consiste à rendre la programmation plus accessible aux débutants, en leur permettant de programmer et de manipuler des représentations visuelles ou tangibles (souvent animées) des concepts de la programmation. Les micromondes incluent les environnements de programmation visuelle et les interfaces de programmation tangible. C'est typiquement le cas de la machine à slots [16], qui fut désignée explicitement comme un micromonde [16]. Elle utilise des commandes symboliques et tangibles du langage LOGO.

Le système Topobo, bien qu'il n'ait pas été à notre connaissance désigné comme un micromonde, présente un grand nombre des caractéristiques de ce dernier. Son concepteur affirme qu'il a été conçu pour permettre à l'utilisateur de lier mentalement des connaissances kinesthésiques avec des connaissances de programmation [17]. Il souligne également certaines similarités avec LOGO : « *Topobo permet d'introduire des idées abstraites, en offrant aux enfants la capacité de simuler les mouvements de leurs réalisations avec les mouvements de leurs propres corps* » [32].

Les environnements Karel [5], Alice [14] et Greenfoot [15] ont été explicitement référencés comme des micromondes dédiés à l'apprentissage de la programmation et de la Programmation Orientée-Objet en particulier [1] [33] [34] [35] [13]. Dans ce qui suit, nous réservons l'expression « *micromondes de programmation* » pour désigner les micromondes dédiés à l'apprentissage de la programmation.

4 Les micromondes de programmation : des systèmes de représentation transitionnels

Nous proposons la formulation de « système de représentation transitionnel », pour désigner le système de représentation propre à un micromonde de programmation [36]. Cette formulation découle de l'expression *objets transitionnels* introduite plus haut.

Les objets transitionnels sont des éléments constitutifs des micromondes de programmation. Ils permettent à l'apprenant d'accéder à des concepts de programmation formels et abstraits, au travers d'entités visuelles et concrètes. Ces dernières sont étroitement liées à ce que l'apprenant connaît de son expérience sensible.

Les objets transitionnels sont choisis de manière à établir un lien sémantique fort entre des concepts formels (ou des objets de science) et des objets concrets du monde réel (ou de l'expérience sensible).

Le système de représentation transitionnel est constitué d'objets transitionnels, visuels et interactifs directement manipulables par l'apprenant. Les objets transitionnels possèdent des propriétés semblables à celles des concepts formels de la programmation et leur aspect externe rappelle des objets du monde réel (Fig. 2).

L'interaction de l'apprenant avec ces objets transitionnels, lui permet de produire du sens sur les concepts de programmation formels, car il est aidé par leurs représentations concrètes et significatives qui sont tirées du monde réel.

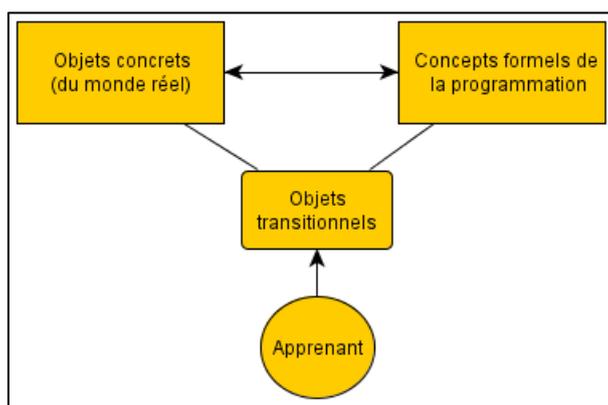


Fig. 2. Système de représentation transitionnel propre aux micromondes de programmation [36].

5 Conclusion

Cet article contribue à l'état de l'art des micromondes de programmation, en décrivant certaines de leurs caractéristiques et en les définissant comme des systèmes de représentation transitionnels.

Les micromondes de programmation sont utilisés afin de favoriser la compréhension des concepts de la programmation. Ce sont des environnements interactifs et restreints, qui contiennent des modèles programmables d'expériences du monde réel. Ils sont engageants, attrayants et permettent un apprentissage fondé sur la participation active des apprenants dans des contextes significatifs.

Ils exploitent les connaissances intuitives de l'apprenant afin de lui permettre d'assimiler et de comprendre des connaissances formelles, au travers de métaphores et au moyen de graphiques ou d'objets physiques. Ils s'étendent au-delà des langages de programmation, afin de permettre à l'apprenant d'explorer et de comprendre le modèle inhérent au paradigme de programmation employé. Cela se fait à travers la manipulation directe d'entités visuelles et concrètes qui constituent les objets transitionnels. Ces derniers étant sémantiquement liés à des concepts de programmation et à des concepts du monde réel, permettent à l'apprenant de construire son apprentissage. L'apprentissage découle des interactions de l'apprenant avec ces objets transitionnels, ce qui constitue le système de représentation transitionnel.

Références

1. J. Bennedsen, «Teaching and learning introductory programming: a model-based approach,» University of Oslo. Faculty of Mathematics & Natural Sciences, 2008.
2. J. Bennedsen, M. Caspersen et M. Kölling, Reflections on the teaching of programming: methods and implementations, Springer, 2008.
3. A. Robins, J. Rountree et N. Rountree, «Learning and teaching programming: A review and discussion,» Computer science education, vol. 13, n°12, pp. 137-172, 2003.
4. A. Mcgettrick, R. Boyle, R. Ibbett, J. Lloyd, G. Lovegrove et K. Mander, «Grand challenges in computing: Education - a summary,» The Computer Journal, vol. 14, n°11, pp. 42-48, 2005.
5. B. W. Becker, «Teaching CS1 with karel the robot in Java,» chez ACM SIGCSE Bulletin, 2001.
6. K. B. Bruce, «Controversy on how to teach CS 1: a discussion on the SIGCSE-members mailing list,» ACM SIGCSE Bulletin, vol. 37, n°12, pp. 111-117, 2005.
7. J. D. Oldham, «What happens after Python in CS1?,» Journal of computing sciences in colleges, vol. 20, n°16, pp. 7-13, 2005.
8. P. Woodworth et W. Dann, «Integrating console and Event-Driven models in CS1,» ACM SIGCSE Bulletin, vol. 31, n°11, pp. 132-135, 1999.
9. B. Moskal, D. Lurie et S. Cooper, «Evaluating the effectiveness of a new instructional approach,» ACM SIGCSE Bulletin, vol. 36, n°11, pp. 75-79, 2004.
10. S. Papert, Mindstorms: Children, computers, and powerful ideas, Basic Books, 1980.
11. J. Bennedsen et C. Schulte, «What does Objects-First mean?: An international study of teachers' perceptions of Objects-First,» chez seventh Baltic Sea Conference on Computing Education Research, 2007.
12. M. Kölling et J. Rosenberg, «Guidelines for teaching object orientation with Java,» ACM SIGCSE Bulletin, vol. 33, n°13, pp. 33-36, 2001.
13. S. Xinogalos, M. Satratzemi et V. Dagdilelis, «An introduction to Object-Oriented Programming with a didactic microworld: objectKarel,» Computers & Education, vol. 47, n°12, pp. 148-171, 2006.
14. S. Cooper, W. Dann et R. Pausch, «Alice: a 3-D tool for introductory programming concepts,» Journal of Computing Sciences in Colleges, vol. 15, n°15, pp. 107-116, 2000.
15. M. Kölling, «The greenfoot programming environment,» ACM Transactions on Computing Education (TOCE), vol. 10, n°14, p. 14, 2010.
16. R. Perlman, «Using computer technology to provide a creative learning environment for preschool children,» ERIC, 1976.
17. H. S. Raffle, A. J. Parkes et H. Ishii, «Topobo: a constructive assembly system with kinetic memory,» chez Proceedings of the SIGCHI conference on human factors in computing systems, 2004.
18. M. Resnick, J. Maloney, A. Monroy-Hernández, N. Rusk, E. Eastmond et K. Brennan, «Scratch: programming for all,» Communications of the ACM, vol. 52, n°111, pp. 60-67, 2009.
19. E. Bruillard, Les machines à enseigner, Hermès, 1997.

20. J. G. Hogle, *Computer Microworlds in Education: Catching Up with Danny Dunn*, ERIC, 1995.
21. R. W. Lawler, «Learning Environments. Now, Then and Someday,» chez *Artificial Intelligence and Education: Learning environments and tutoring systems*, Intellect Books, 1987.
22. L. P. Rieber, «Seriously considering play: Designing interactive learning environments based on the blending of microworlds, simulations, and games,» chez *Educational technology research and development*, vol. 44, Springer, 1996, pp. 43-58.
23. P. Thompson, «Artificial intelligence and instruction: Applications and methods,» chez *Mathematical microworlds and intelligent Computer-Assisted instruction*, JSTOR, 1988.
24. R. W. Lawler et M. Yazdani, *Artificial Intelligence and Education: Learning environments and tutoring systems*, Intellect Books, 1987.
25. T. Winograd, «Understanding natural language,» *Cognitive psychology*, vol. 3, n°11, 1972.
26. M. Minsky et S. Papert, «The 72' Progress Report,» Rapport interne, MIT IA Lab. Publié en 1974, 1972.
27. J. Piaget, «Logique et connaissance scientifique,» chez *Encyclopédie de la Pléiade*, Galimard, 1967.
28. S. Papert, «Microworlds: Transforming education,» chez *Artificial Intelligence and Education: Learning environments and tutoring systems*, Intellect Books, 1987, pp. 79-94.
29. L. P. Rieber, «The Cambridge handbook of multimedia learning,» chez *Multimedia learning in games, simulations, and microworlds*, Cambridge University Press, 2005, pp. 549-567.
30. S. Papert, *The children's machine: Rethinking school in the age of the computer*, Basic books, 1993.
31. M. J. Conway, «Alice: easy-to-learn 3D scripting for novices,» University of Virginia, 1997.
32. H. S. Raffle, «Sculpting behavior: a tangible language for hands-on play and learning,» Massachusetts Institute of Technology, 2008.
33. S. Cooper, W. Dann et R. Pausch, «Teaching objects-first in introductory computer science,» *ACM SIGCSE Bulletin*, vol. 35, n°11, pp. 191-195, 2003.
34. P. Henriksen et M. Kölling, «Greenfoot: combining object visualisation with interaction,» chez *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, 73-82.
35. A. Pears, S. Seidman, L. Malmi, L. Mannila, E. Adams, J. Bennedsen, M. Devlin et J. Paterson, «A survey of literature on the teaching of introductory programming,» *ACM SIGCSE Bulletin*, vol. 39, n°14, pp. 204-223, 2007.
36. F. Djelil, «Conception et évaluation d'un micromonde de Programmation Orientée-Objet fondé sur un jeu de construction et d'animation 3D,» Université Blaise Pascal-Clermont II, 2016.