



HAL
open science

Comprendre la logique de l'exécution d'un programme à l'aide d'interactions tangible

Iza Marfisi, Sébastien George, Marc Leconte, Evan Mottais

► To cite this version:

Iza Marfisi, Sébastien George, Marc Leconte, Evan Mottais. Comprendre la logique de l'exécution d'un programme à l'aide d'interactions tangible. Atelier Apprentissage Instrumenté de l'Informatique, ORPHEE RDV, Jan 2017, Font-Romeu, France. hal-01938575

HAL Id: hal-01938575

<https://hal.science/hal-01938575v1>

Submitted on 28 Nov 2018

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.

Comprendre la logique de l'exécution d'un programme à l'aide d'interactions tangibles

Iza Marfisi-Schottman, Sébastien George, Marc Leconte, Evan Mottais

COMUE UBL, Université du Maine, EA 4023, LIUM, 72085 Le Mans, France
{ Iza.Marfisi, Sebastien.George, Marc.Leconte, Evan.Mottais.etu }@univ-lemans.fr

Résumé : Depuis la réforme de 2016, la programmation informatique est désormais enseignée dès le collège. Les enseignants, qui ne sont pas encore formés dans cette matière, cherchent des outils pédagogiques susceptibles de les aider. C'est dans ce contexte que nous proposons le jeu éducatif TurtleTable qui présente deux originalités majeures. Tout d'abord, cette application n'est pas basée sur des activités d'écriture de programme, comme la plupart des autres outils existants, mais sur l'exécution pas à pas d'un programme donné. Ces deux approches sont complémentaires et nécessaires à l'appréhension de la logique informatique dans sa globalité. Deuxièmement, TurtleTable se joue en manipulant des objets tangibles sur une table interactive. Les joueurs doivent déplacer ces objets sur une grille, en déroulant les instructions du programme qui est affiché à l'écran. Le jeu indique les instructions effectuées correctement en temps réel, permettant ainsi aux apprenants de bien comprendre le principe de l'exécution d'un programme.

Les outils pour l'apprentissage des bases de la programmation

Depuis la rentrée 2016, les principes de la programmation informatique doivent être enseignés au collège. Ainsi, l'informatique et l'algorithmique deviennent des thèmes au programme de mathématiques et technologie. Les enseignants, qui ne sont pas encore formés dans cette matière, ont besoin d'outils pédagogiques susceptibles de les aider. Pour le moment, le site officiel du ministère¹ ne fournit pas beaucoup de ressources, mais il existe de nombreux blogs d'enseignants ou d'associations² qui proposent des fiches d'**activités débranchées**, c'est-à-dire qui peuvent se faire sans ordinateur ni matériel spécifique [1]. On y trouve par exemple le « jeu dont vous êtes le héros » qui se joue à trois : un élève construit un algorithme avec des cartes « avancer », « tourner-droite » et « tourner-gauche », un autre l'exécute en pointant la carte en cours et le dernier élève effectue les déplacements sur un plateau ou un sol carrelé. Les enseignants peuvent aussi se tourner vers les **outils numériques** gratuits pour que les élèves apprennent à programmer à partir de dessin³, d'organigramme⁴, de bloc⁵ ou

¹ <http://eduscol.education.fr>

² <https://pixees.fr/informatique-en-primaire-les-activites-debranchees/>

³ <https://lightbot.com/>

⁴ <http://www.physicsbox.com/demorobotprog.html>

même directement dans un vrai langage⁶. Ces outils numériques ont l'avantage de corriger automatiquement les élèves et de leur proposer plusieurs niveaux de difficulté. Les outils existants proposent tous des activités dont le but est d'écrire un programme pour déplacer un robot (ou un animal) dans un environnement. Les activités sont souvent mises en scène de façon ludique et il est demandé à l'élève de relever des défis de plus en plus difficiles (ex. déplacer le robot afin qu'il récolte tous les bonbons, ne pas utiliser plus de trois lignes de code, etc.). Les concepts de la programmation (variables, conditions, boucles, fonctions) sont introduits progressivement, dans des niveaux de jeux de plus en plus difficiles. Ce type d'activité, basé sur la construction d'un programme, est essentiel à la compréhension de l'algorithmique, mais conduit souvent les élèves à exécuter leur code de façon compulsive et irrationnelle jusqu'à ce qu'il fonctionne. En effet, ce type de jeu n'incite pas les élèves à simuler l'exécution de leur programme à l'avance, pour comprendre comment celui-ci va se dérouler. Il s'agit pourtant d'une des compétences clés pour maîtriser l'algorithmique. Beaucoup d'activité débranchée, comme celle citée ci-dessus, propose d'ailleurs de travailler sur l'exécution d'un programme pas à pas, mais elles nécessitent la présence constante d'un enseignant pour expliquer l'activité, mais aussi pour vérifier que les élèves ne commettent pas d'erreur.

Pour combler ce manque, nous proposons **TurtleTable**, un environnement numérique dont le but est d'entraîner les élèves à exécuter un programme, comme une machine. Il s'agit donc de comprendre comment se déroule l'exécution d'un code informatique et d'être sensibilisé à la logique des instructions. Dans la troisième partie de cet article, nous détaillons comment TurtleTable facilite la collaboration entre élèves avec des interactions tangibles.

Apprendre à exécuter un programme pas à pas

Alors que les outils numériques existants apprennent aux élèves à construire un algorithme, le but de TurtleTable est de leur faire acquérir une compétence complémentaire : savoir exécuter un programme pas à pas. Le principe du jeu est simple : exécuter le programme présenté sur la gauche de l'interface (cadre A, Figure 1) en déplaçant les objets sur la grille (cadre B, Figure 1). Le programme, écrit en pseudo langage, peut contenir des conditions (ex. si $i = 2$), des boucles (ex. pour i allant de 1 à 3) et des instructions simples (ex. déplacer `obj_1` de 3 cases vers l'avant, tourner `obj_2` de 45° vers la gauche). Si l'instruction est exécutée sans faute (la pièce a bien été déplacée au bon endroit), l'instruction se colore en vert et l'élève peut passer à l'instruction suivante. De plus, les déplacements d'objets se matérialisent par un trait sur la grille de façon à créer un dessin. Si les élèves se trompent dans le déplacement de la pièce, ils doivent recommencer l'algorithme au début. L'objectif est de les inciter à prendre le temps de réfléchir et de se concerter avant de déplacer l'objet. Une fois que toutes les instructions sont réussies, les joueurs passent au niveau suivant.

⁵ <https://scratch.mit.edu/>, <https://code.org/learn>

⁶ <http://tortue-logo.fr/fr/tortue-logo>

TurtleTable comporte actuellement 20 niveaux (cadre C, Figure 1) qui introduisent progressivement les concepts basiques de la programmation (i.e. instructions, variables, boucles et conditions). Les joueurs ne manipulent qu'un seul objet dans les premiers niveaux et passent progressivement à deux, puis trois objets.

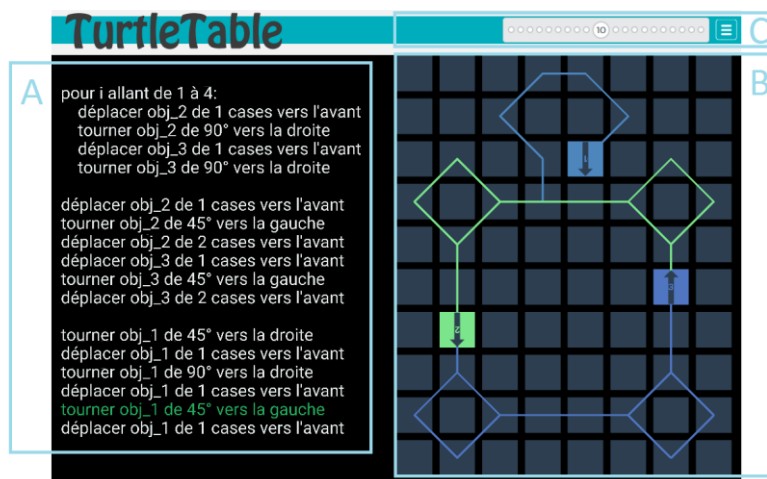


Figure 1. Interface du niveau 10 de TurtleTable

Des interactions tangibles pour faciliter la collaboration

TurtleTable a été conçu pour être utilisé par plusieurs élèves qui doivent collaborer pour résoudre les différents niveaux. Les élèves jouent en manipulant des objets tangibles qui sont reconnus par la table interactive (Figure 2). De nombreuses études ont prouvé que, de par l'engagement corporel et l'invitation à l'action, l'utilisation de tels objets facilitait la collaboration et l'appropriation de concepts, tout en rendant la résolution de problèmes plus ludique et plaisante pour les élèves [3, 4]. D'ailleurs, les fabricants de jouets proposent déjà des jeux d'initiation à l'informatique avec une partie logicielle, installée sur un ordinateur, et des objets tangibles qui communiquent avec ce logiciel. Ces objets sont soit utilisés pour la construction du programme⁷ [2] (pièces qui s'emboîtent) soit pour la construction du robot qui exécute le programme⁸.

L'utilisation d'une table interactive facilite également la collaboration entre équipiers [5]. Cet espace permet aux élèves de visualiser l'interface et d'interagir avec celle-ci de façon équitable, contrairement aux applications sur tablette ou sur ordinateur où, la plupart du temps, une personne interagit et les autres regardent.

Pour finir, certains niveaux de TurtleTable sont conçus pour que chaque élève ait un rôle à jouer en manipulant un objet tangible particulier. Ils doivent également se concerter régulièrement pour exécuter des instructions impliquant plusieurs objets (ex. si obj_1 devant obj_4 ET si mur devant obj_2, alors déplacer obj_3...).

⁷ <http://hci.cs.tufts.edu/tern/>

⁸ <https://www.thymio.org/en:thymio>, <http://www.lego.com/en-us/mindstorms>



Figure 2. Manipulation d'objets tangibles sur une table interactive

Travaux en cours et perspectives

Le premier prototype de *TurtleTable*, qui ne contenait que 2 niveaux et un seul objet tangible, a été testé avec un groupe d'élèves et d'enseignants de collège en juin 2016. Encouragés par le vif intérêt suscité par l'application, nous avons développé le deuxième prototype présenté ci-dessus (avec 20 niveaux et trois objets tangibles) qui sera testé avec une classe entière de 3^{ème} en novembre 2016.

Par la suite, nous envisageons d'ajouter une interface d'édition pour que les enseignants puissent eux-mêmes créer de nouveaux niveaux (paramétrer la taille de la grille, la position initiale des objets et écrire le programme). Il serait également intéressant d'afficher le programme sous différentes formes (dessins, organigrammes, blocs ou différents langages) pour introduire la notion de syntaxe propre à chaque langage et proposer un plus grand éventail de niveau de difficultés.

Références

1. C. Calmet, M. Hirtzig, and D. Wilgenbus, *1, 2, 3... codez ! enseigner l'informatique à l'école et au collège*, Le pommier, 2016.
2. F. Scharf, T. Winkler, and M. Herczeg, "Tangicons: Algorithmic Reasoning in a Collaborative Game for Children in Kindergarten and First Class," in *Proceedings of the International Conference on Interaction Design and Children*, NY, USA, 2008, pp. 242–249.
3. B. Schneider, P. Jermann, G. Zufferey, and P. Dillenbourg, "Benefits of a Tangible Interface for Collaborative Learning and Interaction," *IEEE Transactions on Learning Technologies*, vol. 4, no. 3, 2011, pp. 222–232.
4. S. Price and T. Pontual Falcão, "Where the attention is: Discovery learning in novel tangible environments," *Interacting with Computers*, vol. 23, no. 5, 2011, pp. 499–512.
5. O. Shaer, M. Strait, C. Valdes, H. Wang, T. Feng, M. Lintz, M. Ferreira, C. Grote, K. Tempel, S. Liu, "The design, development, and deployment of a tabletop interface for collaborative exploration of genomic data", *International Journal of Human-Computer Studies*, Volume 70, Issue 10, 2012, pp. 746-764.