

# Apprentissage de la programmation en cycle 2 avec un jeu vidéo collaboratif

Christophe Reffay<sup>1</sup>, Frédéric Dadeau<sup>2</sup>, Bruno Follet-Locatelli<sup>2</sup>,  
Paul-Armand Michaud<sup>2</sup>, Françoise Greffier<sup>1</sup>

<sup>1</sup> Université Bourgogne Franche-Comté, FR-EDUC ELLIAD, 25030 Besançon, France  
{christophe.reffay, francoise.greffier}@univ-fcomte.fr

<sup>2</sup> Université Bourgogne Franche-Comté, FEMTO-ST/DISC, 25030 Besançon, France  
frederic.dadeau@femto-st.fr, {bruno.follet-locatelli, paul-armand.michaud}@edu.univ-fcomte.fr

**Résumé.** Nous présentons dans cet article un jeu vidéo destiné à l'apprentissage collaboratif de la programmation, nommé Collabots. Ce jeu met en scène des robots que deux joueurs doivent programmer pour atteindre un objectif. L'originalité réside dans le fait que le jeu se joue à deux joueurs et nécessite l'interaction entre les robots de chaque joueur pour résoudre les puzzles, obligeant les enfants à communiquer entre eux pour synchroniser leurs actions. Nous proposons un protocole expérimental qui visera à évaluer la pertinence de ce jeu pour l'apprentissage de la programmation dans les classes de primaire.

**Mots-clés.** Jeu vidéo, programmation, collaboration, primaire, évaluation

**Abstract.** We present in this paper a video game, named Collabots, aiming at learning programming in a collaborative way. In this game, the players have to program the robots to achieve a given goal. The originality of this game resides in the fact that it requires the robots to interact in order to solve the puzzle, forcing the children to communicate with each other in order to synchronize their actions. We propose an experimental protocol that will aim to evaluate the relevance of this game in teaching programming in elementary schools.

**Keywords.** Video game, programming, collaboration, elementary school, evaluation

## 1. Introduction

Une expérience locale de conceptions d'activités d'initiation à la programmation à l'école primaire [1], portant autant sur des activités déconnectées [2]–[4] que sur quelques activités connectées, a révélé la nécessité d'un soutien important envers les professeurs des écoles (PE) insuffisamment formés à la « pensée informatique » et aux paradigmes de la programmation. Cette nécessité relève d'un constat général qui a par exemple conduit à la création du MOOC « Class'Code » [5] déployé actuellement en Franche-Comté notamment dans des formations en lien avec le rectorat de Besançon.

Pour avoir visité de nombreux enseignants de primaire, nous avons pu constater un écart important entre la maîtrise actuelle des technologies par les PE, et celle qu'ils pensent nécessaire pour enseigner la pensée informatique. Il convient donc de les

« outiller » en leur proposant des activités déjà didactisées et des applications robustes, simples à mettre en œuvre qui leur donnent envie de poursuivre, pour eux, pour leurs pairs et pour les élèves, l'apprentissage de l'informatique (science) et de la pensée informatique. De là, on peut comprendre la floraison de ressources (fiches d'activités) et d'applications permettant d'explorer ou d'entraîner les élèves dans ce domaine.

C'est dans ce contexte que nous proposons le jeu Collabots, un jeu vidéo qui vise l'apprentissage du codage informatique, à destination des élèves de primaire, en développant la collaboration entre apprenants.

Dans la suite de cette proposition, nous présentons le jeu Collabots que nous avons développé, les éléments de programme du cycle 2 que cet EIAH permet d'exercer, avant de décrire succinctement les objectifs de notre recherche naissante. Ceux-ci pourront trouver ancrage dans le cadre théorique qui suit et qui nous permettra d'affiner nos questions de recherche pour nous permettre d'explicitier un protocole d'expérimentation.

## 2. Description du jeu Collabots

Collabots est un jeu collaboratif destiné à l'apprentissage de la programmation par les enfants dès le cycle 2 (CP-CE2). Dans ce jeu, deux robots se déplacent, chacun sur son damier, case par case, dans l'objectif de rejoindre une case objectif déterminée.

Ce parcours est semé d'embûches, par exemple des barrières électriques empêchent les robots de passer. Celles-ci peuvent être désactivées par des plateformes/interrupteurs, lorsque le robot s'arrête sur celles-ci ou qu'il y dépose un objet. Ainsi, les robots ont la capacité de manipuler des objets, en les ramassant ou en les déposant. Certaines cases du damier (téléporteur) permettent d'échanger un objet d'un damier à l'autre. Le jeu comprend des niveaux de difficulté selon une configuration spécifique de chemin à parcourir, d'objets disponibles, de barrières et de téléporteurs illustrés sur la Figure 1.

Les deux joueurs programment chacun le déplacement de l'un des deux robots en décrivant une suite d'actions. Il y a 6 actions possibles : avancer, réaliser un quart de tour à droite, réaliser un quart de tour à gauche, ramasser un objet ou déposer l'objet transporté, et attendre. Un calcul de score est ajouté pour amener le joueur à trouver le programme optimal : celui qui minimise le nombre d'instructions utilisées.

La difficulté du jeu est progressive. Les premiers niveaux ont pour objectif d'initier chaque joueur avec les commandes de base pour le déplacement. À ce stade, les deux robots progressent de manière indépendante. Les obstacles sur le damier (barrières) sont ensuite introduits. Les premières interactions entre joueurs deviennent nécessaires lorsque les interrupteurs désactivant les barrières d'un damier sont placés sur l'autre damier. Ces niveaux demandent donc aux joueurs de synchroniser leurs actions. Les derniers niveaux, avec les passages d'objets d'un damier à l'autre, vont forcer une réelle collaboration entre les joueurs pour terminer le niveau avec le meilleur score.

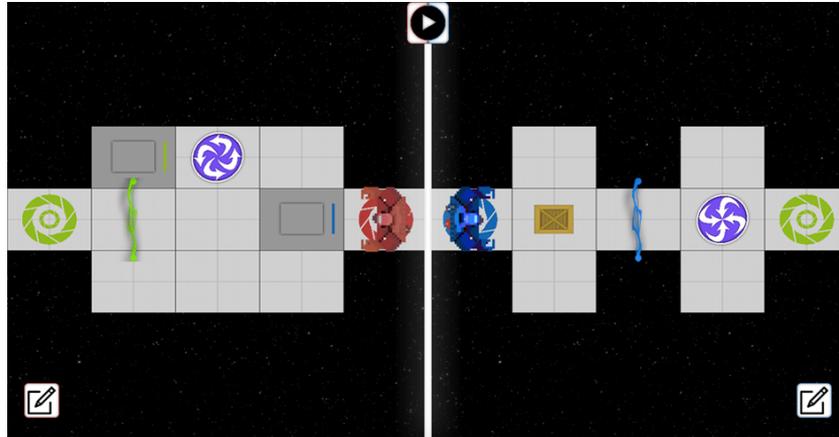


Figure 1. Capture d'écran du jeu Collabots

Un prototype a été réalisé dans le cadre d'un projet d'étudiants de Licence CMI en informatique [6]. Précisons en quoi ce jeu peut contribuer à l'apprentissage de l'algorithmique et en quoi il répond aux programmes du cycle 2.

### 3. Apprentissage de la pensée informatique à l'école

#### 3.1. La pensée informatique à l'école

Nous tentons de synthétiser la définition de la pensée informatique et de l'informatique vue comme une science, avant de présenter ce que retiennent les programmes au cycle 2.

L'excellent billet [7] dans [8] définit l'informatique comme « l'étude du calcul, c'est-à-dire de ce qui peut être calculé mécaniquement. » en explicitant ce qui caractérise la pensée informatique. Mais c'est dans [9] que l'on trouve la meilleure compilation des définitions de l'informatique. On retiendra en particulier les suivantes. [10] proposent des principes scientifiques fondamentaux de calcul, à partir desquels ils interprètent « l'informatique comme l'étude des propriétés fondamentales des processus d'information, aussi bien naturels qu'artificiels où les ordinateurs sont des outils et non des objets d'étude et où le calcul est omniprésent dans la vie quotidienne ». Selon [9], « cette doctrine révèle aussi qu'il existe quelque chose d'encore plus fondamental que l'algorithme : la représentation, qui porte en elle de l'information ». Tandis que dans [11], la science informatique est définie par ses quatre composants fondamentaux et indissociables que sont les concepts d'algorithme, de machine, de langage et d'information.

Au cœur de la pensée informatique se trouvent bien l'algorithme qui peut suffire, sur le plan intellectuel à caractériser la solution à un problème ou à décrire un processus (ex : une recette de cuisine ou un chemin dans un environnement discret). Selon [12], « Un algorithme est un enchaînement mécanique d'actions, dans un

certain ordre, qui chacune a un effet, et dont l'exécution complète permet de résoudre un problème ou de faire quelque chose. »

Au-delà de la communauté francophone, nous trouvons des travaux, des réflexions et des expériences extrêmement intéressantes issues de la conférence ISSEP : (Informatics in Schools: Situation, Evolution and Perspectives) de 2005 à 2016. Ces actes présentent des situations à l'école primaire, ayant un lien avec notre approche par un jeu de déplacement de robots via un simulateur, ou plus largement des travaux sur l'introduction de la pensée informatique à l'école primaire. [13] rapporte une longue expérience de l'enseignement en Suisse de l'algorithmique et de la programmation avec Logo et nous montre avec le recul, toute l'importance de démarrer cet apprentissage dès le plus jeune âge pour développer un curriculum en spirale autour de la pensée informatique. [14] nous offre une excellente contributions selon deux aspects : une forme originale d'informatique débranchée sur de très nombreux sujets informatiques (System d'exploitation, traitement d'image, cryptographie, tri par insertion, théorie des automates, etc.) et une évaluation de leurs interventions (d'enseignants universitaires dans des écoles primaires, collèges et lycées) sur un panel de 18 classes, à travers 5 instruments de recherche permettant d'évaluer les conceptions, leur intérêt et leur perception de l'informatique (avant et après l'intervention), leur attitude vis-à-vis de l'intervention, l'envie que ce type d'interventions procure aux enseignants de prendre en charge ce type d'enseignement ainsi que les différentes représentations de genre (intérêt, capacités) vis-à-vis de l'informatique.

La communauté internationale produisant les concours nationaux d'informatique pour les enfants (Bebras, Beaver, Castor, ...) publie régulièrement des travaux d'intérêt dans ISSEP. Citons, Valentina Dagiené (Lithuanie) et Françoise Tort qui a créé le Concours Castor informatique en France [15]. Des exemples de tâches issues de cette communauté sont proposées comme ressources pour l'enseignement dans [16].

Toutes ces idées et proposition de partage, devraient offrir une meilleure place à l'informatique en tant que science à l'école primaire : « Ainsi nous pourrons communiquer la joie, l'enthousiasme et le dynamisme de l'informatique, dans le but de diffuser largement la pensée informatique » [7].

### **3.2. La place de la pensée informatique dans les programmes au primaire**

Un extrait du BO de nov. 2015, proposé par Class'Code, explicite les notions en rapport avec les connaissances et compétences que Collabots permet d'acquérir ou d'entraîner. Concernant le cycle 2 (volet 3), les préconisations suivantes sont mentionnées. Les enseignements du domaine de l'espace et la géométrie visent la compétence suivante : « Coder et décoder pour prévoir, représenter et réaliser des déplacements dans des espaces familiers, sur un quadrillage, sur un écran », avec, comme situations de référence : « Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran ». Enfin, dans les repères de progressivité on peut lire : « Dès le CE1, les élèves peuvent coder des déplacements à l'aide d'un logiciel de programmation adapté, ce qui les amènera au CE2 à la compréhension, et la

production d'algorithmes simples. » Collabots, tout comme d'autres jeux (Lightbot, Code Studio, etc.) visent cet objectif.

#### **4. Cadre théorique**

La formation à la programmation pour les enfants comme pour les formateurs se base en général sur des activités concrètes, qui visent une appropriation des outils et de savoir-faire. Dans l'apprentissage de la programmation, cette approche expérientielle bien ancrée dans les communautés d'enseignants est en accord avec le constructivisme de Piaget qui met en avant l'activité de l'apprenant et surtout le fait que les connaissances nouvelles se construisent à partir des connaissances acquises antérieurement. Seymour Papert (conférencier invité à EIAH2003 à Strasbourg) est quant à lui considéré comme le père du constructionnisme (comme théorie d'apprentissage) qui prolonge le constructivisme en prônant la manipulation physique d'objets tangibles (comme des robots).

Paradoxalement (les tablettes et ordinateurs étant plus chers que les mini-robots, mais destinés à de nombreux usages possibles dans presque toutes les disciplines), il est plus réaliste d'envisager un écran de tablette ou d'ordinateur par élève dans les classes. Ainsi, on trouve de nombreux simulateurs de programmation de robots pour l'apprentissage de la programmation en classe de primaire ou en collège [17]–[20].

Devant une telle offre, pourquoi proposer un nouveau jeu ? La proposition faite à travers le jeu Collabots, est originale sur deux points :

1. Les instructions de base (actions pour le robot) sont plus nombreuses et dépendent du contexte dans lequel se trouve le robot (ex : prendre une caisse, attendre, etc.), invitant ainsi l'utilisateur à explorer le monde pour en comprendre les pièges et les possibilités offertes avant de construire une stratégie pour réussir (essai-erreur). Il oblige/invite également le joueur à optimiser le nombre d'actions.
2. Le jeu est conçu pour deux joueurs. Il invite, voire oblige les enfants à la collaboration (patience, aide, explications, synchronisation, etc.)

Ainsi, si le jeu Collabots partage, avec les robots et les simulateurs, les objectifs de l'apprentissage de la programmation pour les enfants du plus jeune âge (cycle 2), sa principale particularité réside dans le fait qu'il se joue à deux et demande aux joueurs de collaborer.

C'est à partir de cette spécificité que nous souhaitons mettre en place un dispositif de recherche-action pour évaluer l'impact de la collaboration dans Collabots sur l'efficacité de l'apprentissage de la programmation chez les enfants de cycle 2.

#### **5. Description du projet de recherche**

Dans cette partie, après avoir écarté des questions d'IHM ou purement liées au développement du jeu, nous avons décidé de centrer notre étude sur l'évaluation des

apprentissages que permet le jeu Collabots. Nous présentons dans cette partie les grandes lignes de l'évaluation que nous comptons mener dans une classe-test.

### 5.1. Questions liées à l'apprentissage

La collaboration en binôme sur une seule tablette améliore-t-elle l'efficacité de l'apprentissage de l'algorithmique par le jeu, pour des élèves de cycle 2 ? Plus précisément, quelle articulation (frein ou atout) observe-t-on entre la nécessité de mobiliser des capacités de négociations (savoir-être) et l'intégration des savoir-faire liés à la compréhension d'algorithmes ? Sachant par exemple, que des manipulations par essai-erreur sont possibles, dans quelle mesure cette stratégie sera adoptée plutôt que la concertation à deux ?

### 5.2. Questions de recherche

Peut-on mesurer ce que le dispositif Collabots permet aux enfants d'apprendre ?

Partant de là, il nous faut expliciter les apprentissages à évaluer, concevoir le dispositif expérimental, l'échantillon, les instruments de collecte des données, et définir nos outils d'analyses. Une première ébauche des questions qui pilotent notre réflexion est proposée ci-dessous.

- Quelles sont les pré-requis pour pouvoir utiliser le dispositif ?

Il s'agit ici d'identifier les savoirs et savoir-faire nécessaires pour comprendre l'objectif de chaque niveau du jeu (que chaque robot atteigne sa case cible). Également, il faudra évaluer les concepts à mobiliser pour atteindre cet objectif.

- Quels sont les concepts, connaissances/compétences visés par le dispositif ?

Ce dispositif s'appuie sur la programmation d'une séquence d'instructions qui sont les 6 actions possibles du robot. Il est attendu que l'élève soit capable de produire la séquence minimale d'actions permettant d'atteindre l'objectif. Un calcul de score basé sur cette métrique l'y encourage. Il est également attendu que les élèves soient capables de synchroniser les actions de leurs robots, éventuellement en les temporisant (action « attendre ») pour permettre la résolution du niveau considéré, notamment lorsqu'un robot doit se placer sur un interrupteur le temps que l'autre franchisse une barrière, ou lorsqu'un robot doit envoyer un objet à l'autre qui doit le réceptionner.

- Quels sont les concepts, connaissances/compétences acquises suite à l'utilisation du dispositif ?

Il s'agira ici d'évaluer si le jeu Collabots remplit l'objectif pédagogique fixé dans les programmes scolaires.

- Quel est l'intérêt de l'aspect collaboratif du jeu ? Celui-ci apporte-t-il un bénéfice à l'apprentissage ? Comment s'observe-t-il, se mesure-t-il, se quantifie-t-il ?

Nous avons constaté lors de nos expérimentations en classe de primaire que beaucoup d'enfants cherchent naturellement à aider leurs camarades lorsque ceux-ci bloquent sur une difficulté, même si l'exercice demandé est à effectuer seul. Par ailleurs, certains enfants, notamment dans les classes d'ULIS ont parfois des difficultés de

communication avec les autres [1]. Si l'aspect collaboratif de Collabots n'est (potentiellement) pas central pour l'apprentissage de la programmation de robots en soi, nous parions qu'il peut avoir une influence positive du point de vue des interactions humaines pour faciliter l'apprentissage, en particulier à destination des enfants en difficultés ou en situation de handicap.

### 5.3. Dispositif expérimental envisagé

Nous prévoyons ainsi de concevoir un dispositif expérimental, permettant de répondre à ces différentes questions de recherche. Nous en donnons ci-après les grandes lignes.

Celui-ci commencera par un pré-test (sur les prérequis, les compétences et connaissances visées), puis nous évaluerons le jeu avec différents groupes (groupe test avec le jeu Collabots en duo, groupe témoin avec le jeu en solo). Nous mesurerons ensuite par un post-test les acquis d'apprentissage. Cette expérimentation sera proposée dans une ou plusieurs classes que nous espérons représentatives du public visé. Nous prévoyons également de solliciter des unités localisées pour l'inclusion scolaire (ULIS) auxquelles nous pouvons avoir accès.

Pour nous permettre cette évaluation expérimentale, nous nous appuyerons sur les variables de données suivantes : le contexte de l'expérience (école, classe, avancement dans le programme, notions déjà rencontrées, caractéristiques de l'enseignant, etc.), les individus considérés (âge, sexe, handicap, etc.), la trace des actions de chaque joueur via une instrumentation du jeu Collabots, une capture vidéo des interactions (verbales, para-verbales, co-verbales), l'enregistrement et la transcription des dialogues.

Ces expérimentations seront menées, suivant le calendrier de déploiement du jeu, en fin d'année scolaire 2016-2017 ou en début d'année scolaire suivante.

## 6. Conclusion

Nous avons présenté dans cet article le jeu vidéo Collabots, développé à l'université de Franche-Comté. Celui-ci vise l'apprentissage collaboratif du code informatique, par la programmation de robots virtuels, qui doivent atteindre un objectif précis de ce *puzzle-game*, selon plusieurs niveaux de difficultés. Le développement de Collabots convoque plusieurs plans : (i) penser la conception d'un outil à livrer clé en main auprès des PE ; (ii) niveau didactique : proposer un jeu pour des élèves de cycle 2, dédié à l'apprentissage de programmes de parcours de robots (niveaux de difficultés croissants) ; (iii) associer à ce but didactique, la nécessité de mettre en œuvre des capacités collaboratives.

Nous avons proposé un ensemble de questions liées à l'apprentissage ainsi que des questions de recherche liées à l'application de ce jeu avec les élèves. La prochaine étape visée est l'expérimentation du jeu dans une classe. Auparavant, nous procéderons à la finalisation du jeu, permettant de nous assurer de son ergonomie, sa facilité d'utilisation, et l'absence de bugs, qui sont des éléments rédhibitoires à son adoption par le public visé (élèves de l'école élémentaire). Nous prévoyons aussi le

déploiement du jeu sur tablette, ce qui nous permettra ensuite de réaliser une immersion dans une classe-test, avec une captation vidéo de la séance. A ces fins, nous prévoyons également l'instrumentation du jeu pour permettre d'enregistrer les programmes successifs proposés par les élèves, et ainsi analyser leurs tentatives.

## 7. Références

Tous les liens de cette bibliographie ont été vérifiés en Avril 2017.

1. A. Alavoine et F. Dadeau, « Introduction du code informatique à l'école primaire - retour d'expériences », 1014 Bull. Société Inform. Fr., vol. 9, p. 61-71, nov. (2016).
2. T. Bell, J. Alexander, I. Freeman, et M. Grimley, « Computer science unplugged: School students doing real computing without computers », N. Z. J. Appl. Comput. Inf. Technol., vol. 13, no 1, p. 20-29, (2009).
3. T. Nishida, Y. Idosaka, Y. Hofuku, S. Kanemune, et Y. Kuno, « New Methodology of Information Education with "Computer Science Unplugged" », in Informatics Education - Supporting Computational Thinking: ISSEP 2008 Torun Poland, July 1-4, 2008 Proceedings, R. T. Mittermeir et M. M. Sysło, Éd. Springer Heidelberg, p. 241-252. (2008).
4. « Computer Science Unplugged », Disponible sur : <http://csunplugged.org/>. (2017).
5. « Class'Code – Maîtriser la pensée informatique pour la transmettre », (2017).
6. B. Follet et P.-A. Michaud, « Apprentissage de l'informatique et de la programmation à l'école », Département d'informatique, Université de Franche-Comté, Besançon, Projet L3 CMI Informatique, (2017).
7. J. Jongwane, « La pensée informatique », Interstices, 29-mai-2009. Disponible sur : <https://interstices.info/pensee-informatique>. (2009).
8. « Interstices ». Disponible sur: [https://interstices.info/jcms/jalios\\_5127/accueil](https://interstices.info/jcms/jalios_5127/accueil). (2017).
9. M. Grandbastien, « La discipline informatique et l'éducation à la maîtrise de l'information », p. 12, janv. (2012).
10. D. E. Comer, D. Gries, M. C. Mulder, A. Tucker, A. J. Turner, et P. R. Young, « Computing As a Discipline », Commun ACM, vol. 32, no 1, p. 9-23, janv. (1989).
11. G. Dowek et G. Berry, Une introduction à la science informatique pour les enseignants de la discipline en lycée. Paris: Centre régional de documentation pédagogique de l'académie de Paris, (2011).
12. P. Tchounikine, « Initier les élèves à la pensée informatique et à la programmation avec Scratch ». (2016).
13. G. Serafini, « Teaching Programming at Primary Schools: Visions, Experiences, and Long-Term Research Prospects », in Informatics in Schools. Contributing to 21st Century Education: ISSEP 2011, Bratislava, Slovakia, October 26-29, 2011. Proceedings, I. Kalaš et R. T. Mittermeir, Éd. Springer Berlin Heidelberg, p. 143-154. (2011).
14. E. Bischof et B. Sabitzer, « Computer Science in Primary Schools – Not Possible, But Necessary?! », in Informatics in Schools. Contributing to 21st Century Education: ISSEP 2011, Bratislava, Slovakia, October 26-29, 2011. Proceedings, I. Kalaš et R. T. Mittermeir, Éd. Springer Berlin Heidelberg, p. 94-105. (2011).
15. F. Tort, « Le concours Castor : un outil de promotion de l'enseignement d'informatique », présenté à Sciences et technologies de l'information et de la communication en milieu éducatif : Analyse de pratiques et enjeux didactiques, p. 221-230. (2011).
16. V. Dagienė et S. Sentance, « It's Computational Thinking! Bebras Tasks in the Curriculum », in Informatics in Schools: Improvement of Informatics Knowledge and Perception: ISSEP

- 2016, Münster, Germany, October 13-15, 2016, Proceedings, A. Brodник et F. Tort, Éd. Springer International Publishing, p. 28-39. (2016).
- 17.« Le Campus Junior », 2017. Disponible sur : <https://www.lecampusjunior.fr/>
- 18.« Pour les petits génies de primaires | Code Moi Un Mouton ». Disponible sur : <http://codemoiunmouton.fr/?p=118>. (2017).
- 19.« Lightbot », 2017. Disponible sur : <https://lightbot.com/index.html>.
- 20.« Apprendre à coder et programmer », Geek Junior, 15-juill-2016. Disponible sur : <https://www.geekjunior.fr/apprendre-a-coder-programmer-50-ressources-enfant-ados-4376/>