

Initiation à la pensée informatique avec le jeu de plateau Programming Boty

Lydie Boufflers, Sophie Linh Quang, Daniel K. Schneider

Université de Genève, FPSE / TECFA, 1205 Genève, Suisse
lydie.boufflers@etu.unige.ch, sophie.linh@etu.unige.ch, daniel.schneider@unige.ch

Résumé. Le kit “Programming Boty”, un jeu de plateau, a pour objectif de proposer une initiation à la programmation et à l’algorithmique aux enfants de l’école primaire, et plus généralement aux novices de l’informatique. Ce design *open source* réalisé avec une découpeuse laser s’inspire de la pédagogie Montessori qui stipule que la manipulation d’objets physiques favorise l’appropriation de concepts abstraits. Cet article présentera la méthode de conception et de réalisation, le kit d’apprentissage, et les résultats obtenus au test utilisateur effectué.

Mots-clés. Programmation, éducation, jeu de plateau, découpe laser, objet tangible

Abstract. The kit "Programming Boty", a board game, provides an introduction to programming and algorithmic to primary school children and more generally to computer science novices and was made with a laser cutter. The design of this kit is based on Montessori pedagogy, which stipulates that the manipulation of physical objects favors the appropriation of abstract concepts. This article will present the design method and process, the kit, and the results of user test

Keywords. Programming education, board game, laser cutting, tangible object

1. Introduction

Pour préparer les élèves à la société informatisée l’école devrait enseigner les bases de l’informatique [12]. Apprendre les bases de l’algorithmique et de la programmation contribue à développer des compétences transversales et fondamentales. Ils sont indispensables pour mieux comprendre les TIC, être capable d’en résoudre des problèmes et garder un esprit critique vis-à-vis des dispositifs qui nous sont proposés.

Plusieurs approches existent pour enseigner les bases de la programmation et de l’algorithmique [4]. On peut d’abord distinguer les environnements de programmation pour novices que Guzdial [5] classifie en trois familles: les micro-mondes exploratoires (Logo et descendants), les systèmes basés sur des règles et les environnements de support pour les langages traditionnels. Dans un registre différent, il existe des jeux et autres artefacts qui visent à enseigner des principes de l’informatique sans utiliser l’ordinateur. Un bon exemple est le projet “Computer Science Unplugged” [6] [7].

Programming Boty, un jeu de plateau, permet d’enseigner quatre concepts fondamentaux de la programmation: le programme, l’instruction, les procédures et

l'appel à des procédures. Ce kit s'inspire de plusieurs sources: La première idée est venue du jeu Lightbot [1], lui-même inspiré du jeu en ligne Bill the Robot [2]. Ensuite, on retrouve l'héritage de la tortue de Logo [9]. Sur le plan théorique, le kit appartient à une catégorie d'objets d'apprentissage nommée "manipulation conceptuelle" par Zuckerman [11]: en manipulant un objet physique, les apprenants parviennent à «absorber» un concept abstrait. Nous avons aussi favorisé l'autonomie en y ajoutant une dimension d'apprentissage collaboratif, des principes que l'on retrouve dans le constructionnisme véhiculé par les micro-mondes de programmation [9].

Utilisé dans le contexte scolaire, ce dispositif est plus économique et plus solide que des versions numériques dont il s'inspire. On présuppose aussi qu'un jeu de plateau favorise la collaboration entre enfants puisque les manipulables sont faciles d'accès. Dans cet article seront détaillés des éléments de la conception, le kit, le test utilisateur et ses résultats, pour terminer avec une discussion.

2. Cahier des charges et méthode

Objectifs. Programming Boty est un kit de construction destiné à proposer une initiation à la programmation afin d'initier à la pensée informatique. A l'issue de l'apprentissage avec ce kit, les apprenants seront capables d'utiliser les principes de base de la programmation et de l'algorithmique en manipulant des instructions (avancer, tourner, prendre, appeler une procédure) et des procédures. Le kit est destiné aux classes de degré primaire (enfants de 6 à 12 ans) où l'enseignant interviendrait pour expliquer les règles du jeu et gérer le niveau de difficultés (niveau 1, 2 ou 3). Ce kit peut également être destiné à des adultes, novices en algorithmique et programmation. Un modérateur jouerait alors le rôle que l'enseignant occupe pour les élèves de degré primaire.

Exigences et contraintes de conception : le kit doit favoriser l'autonomie de l'apprentissage (Pédagogie Montessori), une pédagogie active et l'apprentissage par découverte. Il doit également apporter une dimension collaborative, une dynamique de groupe. Le jeu doit être accessible aux novices et adaptable à plusieurs profils d'individus (enfants, seniors...). Il ne doit être ni infantilisant ni trop simpliste. Le jeu doit être compatible avec un enseignement en classe. Le fait que l'enseignant n'aura pas le temps de s'occuper simultanément de tous les groupes favorise également la pédagogie active que nous souhaitons mettre en place à travers ce kit.

Méthode de développement et réalisation. La réalisation du kit s'est divisée en plusieurs phases. Dans un premier temps un prototype papier a été réalisé afin de matérialiser les dynamiques et mécaniques de jeu et d'expérimenter au sein du groupe de développement, composé des deux premiers auteurs. Dans une seconde étape les designs des différents éléments SVG composant le kit ont été réalisés avec le logiciel Inkscape. Dans un troisième temps, un prototype carton créé avec une découpeuse laser a permis d'effectuer un test utilisateur. La réalisation finale fut exécutée sur contreplaqué bouleau 4mm, également réalisé avec la découpeuse laser.

Tests utilisateurs. Un test utilisateur avec le prototype carton a été effectué sur un groupe de 4 personnes, composé de 2 hommes et 2 femmes. Les sujets sont des habitués de jeux de société pour lesquels ils se réunissent régulièrement. Les participants ont été informés du test à réaliser : *jouer à un jeu collaboratif qui a pour objectif d'enseigner*

les bases fondamentales de la programmation. On a précisé qu'il s'agissait de tester le matériel et non pas leurs capacités à réussir et que les données personnelles resteront confidentielles. Puis, l'expérimentatrice a procédé à la mise en place du jeu, en a expliqué les règles tout en présentant les différents éléments composant le kit. Elle a ensuite mis en place la première map de positionnement. Tout au long de la séance, l'expérimentatrice, en prenant le rôle "d'enseignante" dispose les objets et le robot sur le plateau selon les maps prédéfinies.

A la fin de la séance, lorsque les joueurs ont terminé les 14 maps proposées, on a évalué le kit. Tout d'abord, un entretien ouvert s'est engagé à propos du kit avec des questions du type : « Qu'avez-vous pensé du jeu? » « Selon vous, quels sont les points forts et les points faibles du jeu? » puis, les participants ont été soumis à une version adaptée du questionnaire SUS (System Usability Scale) [10].

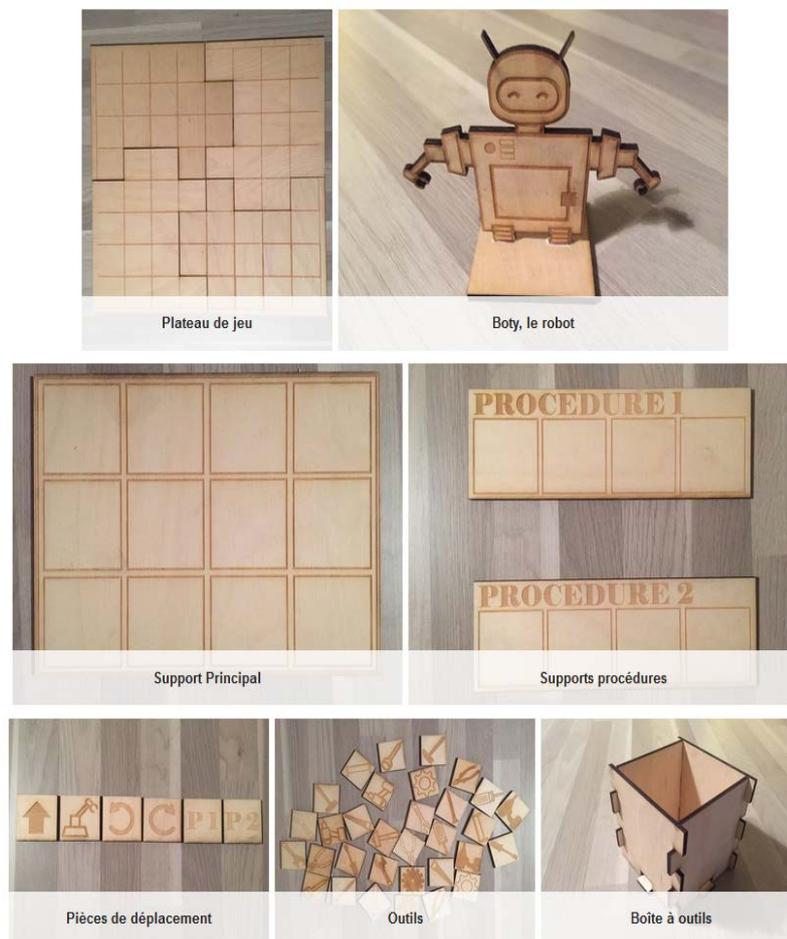


Fig 1. Photographie des divers éléments composant le jeu

Ils devaient indiquer leur degré d'accord avec 10 propositions dont la moitié est formulée positivement, l'autre moitié négativement. Chaque proposition est évaluée à

l'aide d'une échelle de Likert en 5 points, de 1 (pas du tout d'accord) à 5 (tout à fait d'accord). La somme des réponses à chaque questionnaire est multipliée par 2.5. Cela permet d'obtenir un score d'utilisabilité subjective allant de 0 à 100.

3. Réalisation

Le dispositif est composé d'un kit physique, également disponible sous forme de fichiers SVG en ligne, et d'un manuel d'utilisation. Le kit final est composé de:

1. Un plateau de jeu ;
2. Boty, le robot à déplacer ;
3. des "jetons de déplacements", des instructions pour programmer les mouvements de Boty sur le plateau ;
4. des "jetons outils" (pièces mécaniques) que Boty doit ramasser au cours de son parcours ;
5. des "supports principaux" pour la disposition des différentes instructions (i.e jetons de déplacements) permettant de mouvoir Boty ;
6. des "supports procédures" pour la conception des procédures par les joueurs (niveau 2 et 3) ;
7. un livret contenant les [règles du jeu](#) et les maps

Avant de commencer, le robot et les pièces mécaniques sont placés sur le plateau. Pour cela, des "maps" de positionnements sont à disposition selon différents degrés d'expertises que l'enseignant / modérateur introduit petit à petit:

1. dans un premier temps, des déplacements simples en n'utilisant que le support principal (niveau 1) ;
2. puis introduction de la procédure 1 (niveau 2) ;
3. et enfin l'utilisation des deux procédures à la fois (niveau 3).

L'objectif du jeu est de programmer les déplacements du robot afin que ce dernier aille ramasser toutes les pièces mécaniques disposées sur le plateau. La difficulté du jeu augmente donc avec le nombre des pièces disposées sur le plateau et le nombre limité de pièces de déplacements pouvant être positionnées sur le support principal (d'où la nécessité pour les joueurs de recourir aux supports procédures pour pouvoir résoudre l'énigme). Le jeu standard se déroule en deux étapes:

1. Phase de réflexion : les joueurs réfléchissent à une solution et placent les jetons de déplacement sur leurs supports pour organiser leur solution (support principal et supports procédures);
2. Phase d'action : les joueurs tentent de résoudre l'énigme en déplaçant le robot selon leur programmation. Par exemple, on peut imaginer que l'un lit le code pendant que l'autre met le robot en mouvement par exemple. Peut-être aussi l'enseignant peut entrer en jeu à ce moment-là, pour vérifier qu'ils exécutent correctement les déplacements.

A cette étape, soit la programmation est réussie (le robot a réussi à ramasser toutes les pièces), soit la programmation comporte encore des erreurs. Dans ce dernier cas, les joueurs reprennent l'étape de réflexion, et ainsi de suite jusqu'à ce qu'ils trouvent une solution.

Afin que le jeu se prolonge au-delà de l'apprentissage de concepts nous avons aussi conçu une utilisation en deux modes: Le mode "practice" correspond à la phase d'apprentissage des concepts de programmation. L'apprenant est seul ou en groupe (de 2 à 5 joueurs). En groupe, ils doivent réfléchir collectivement à la combinaison de variables la plus efficace possible afin que le robot puisse atteindre son but en utilisant un minimum de variables. Pour cela, ils doivent échanger et se mettre d'accord sur la procédure en classant les éléments sur le support mais sans pouvoir toucher physiquement le robot sur le plateau. Ils doivent donc être capables de conceptualiser la procédure mentalement avant de la réaliser physiquement sur le plateau.

Le Mode compétition permet de prolonger l'utilisation de l'outil pédagogique (apprentissage de concepts) décrit dans le mode "practice". Il fait intervenir deux groupes d'apprenants qui se livrent une 'battle' pour trouver le plus rapidement possible la solution. Également, pour apporter de l'intérêt au jeu et surtout susciter la motivation des apprenants, un système de points a été imaginé pour le mode compétition.

6 Résultats des tests d'utilisation

Observation et verbatims. Même si les novices ont d'abord montré quelques signes d'appréhension, notamment une certaine crainte vis-à-vis du mot "programmation" en tout début de séance, l'ensemble des participants semble avoir rapidement compris les règles du jeu. Durant le jeu, les joueurs ont donné chacun leur opinion et ils échangent beaucoup ce qui semble refléter une bonne dynamique de groupe et une bonne coopération. Le joueur "expert" laisse les autres joueurs manipuler les pièces et se place davantage en observateur. Il intervient seulement lorsque les autres joueurs semblent se trouver dans une impasse en donnant des conseils, tels que: "là on pourrait essayer de mettre la P1 dans la P2". Au bout de quelques maps, les novices ont commencé à élaborer des stratégies. En effet l'un d'entre eux s'exprime ainsi: "*il faut regarder les éléments qui se répètent, comme ça on peut les mettre en procédure*".

Problèmes observés : (1) En phase de résolution: les apprenants lisent la solution et déplacent le robot, mais ils ne voient pas leurs erreurs entre la lecture et le déplacement effectif (par exemple: ils lisent "tourner à gauche" (ce qui est la bonne solution) alors qu'ils ont placé un "tourner à droite" sur le support). (2) Les joueurs trouvent qu'il y a trop de niveaux faciles.

Propositions de remédiation : (ad 1) Pour le mode practice: c'est l'enseignant qui lit la solution et la contrôle. Pour le mode compétition: c'est l'équipe adverse qui lit la solution et la contrôle. (ad 2) On peut adapter le nombre de niveaux faciles suivant la population: les adultes comprennent rapidement les règles de déplacement simples, ils n'ont pas besoin d'autant de maps de niveau 1 et 2. Pour eux, il faut peut-être seulement prévoir 1 map de positionnement de démonstration (l'enseignant ou modérateur montre aux joueurs le fonctionnement), 1 map de niveau 1, puis 1 map de niveau 2 pour enfin concentrer le jeu sur les maps de niveau 3, puis éventuellement en développer d'autres.

Entretiens ouverts : Deux questions ouvertes ont été posées aux participants. Le but de la première, "*Qu'avez-vous pensé du jeu lorsque vous effectuiez les tâches demandés?*" était de mieux comprendre ce qu'avait pu ressentir l'utilisateur pendant l'utilisation du kit. *Les participants ont tous trouvé que l'activité était très ludique. Ils*

déclarent tous s'être beaucoup amusés. Ils étaient tous d'accord pour dire que la prise en main a été simple et très rapide. Ils ont aimé l'aspect collaboratif et ont estimé que le jeu constituait une bonne activité de groupe. Les participants pensent que ça pourrait être une activité également "sympa" et adaptée pour les enfants. A ce propos, l'expert parle de son expérience d'apprendre la robotique à des enfants de 8 ans. Il déclare qu'il s'agit souvent d'une activité très intuitive pour eux, et qu'ils s'en sortent souvent mieux que les adultes. Ces réponses obtenues sont globalement très positives et elles ont tendance à souligner des aspects qui leur ont semblés positifs, tels que l'aspect ludique du kit d'apprentissage, la prise en main rapide, l'aspect collaboratif ou encore l'adaptabilité du kit pour différents publics.

La seconde question, "*Selon vous, quels sont les points forts et les points faibles de ce jeu?*", avait un objectif plus contrasté et a permis de faire ressortir d'autres aspects, tels que les limites d'un tel jeu. Les points positifs sont les suivants : *Univers robotique. La difficulté progressive permet de bien faire comprendre les concepts. Le personnage-robot est sympathique. L'idée d'un jeu collaboratif. La possibilité d'ajouter des extensions.* Les points faibles : *On ne sait pas où mettre les pièces une fois ramassées.* De ce constat émerge la proposition de faire une "boîte à outils". *La question de la longévité du jeu (nécessite beaucoup de maps de positionnement).* Cependant, rien n'empêche l'enseignant / modérateur voire même les participants de créer de nouvelles maps comme il l'a été mentionné dans le document règles du jeu. *Un des joueurs déclare qu'il aurait préféré un mode de jeu plus compétitif, cela lui procurerait davantage de motivation* (sachant que le groupe n'a pu expérimenter que la version "practice").

Scores obtenus au System Usability Scale (SUS) adapté : Le score moyen obtenu au questionnaire SUS est très bon (85/100). Aux vues des commentaires faits par les utilisateurs lors de l'entretien ouvert, ce score est cohérent. Les utilisateurs sont, en effet, globalement satisfait du jeu et lui trouvent des points forts significatifs. Cependant, quelques axes d'amélioration plus mineurs ont été soulevés. Ce score, mis en relation avec les commentaires des utilisateurs permet d'avoir une évaluation quantitative aux commentaires qualitatifs faits lors de l'entretien ouvert.

7. Discussion

L'objectif de ce travail était de réaliser un kit de construction destiné à initiation à la programmation informatique et à l'algorithmique. Aux vues du résultat final en terme matériel et des commentaires positifs des utilisateurs, nous pouvons conclure que la solution choisie répond à l'objectif de départ d'un point de vue utilisabilité.

Néanmoins il est nécessaire de prendre en compte quelques limites du test utilisateurs. En effet, le test a été effectué sur seulement quatre individus adultes et, pour des raisons techniques, le test n'a pu s'effectuer que sur le mode practice et pas sur le mode compétition. En effet, le kit n'étant qu'au stade de prototype carton au moment du test, le nombre de pièces à disposition était limité et ne permettait ainsi pas de tester le mode compétition. Un autre biais à souligner est que les participants étaient des joueurs réguliers de jeu de société, donc probablement habitués à apprendre rapidement de nouvelles règles de jeu. Par ailleurs, nous aurions pu penser que ces

derniers auraient pu être très exigeants en matière de support ludique du fait de leur expérience.

Les résultats montrent une opinion très positive envers le kit proposé. Ainsi, dans le prolongement de ce projet, il conviendrait de réaliser d'autres tests utilisateurs et notamment auprès des enfants qui constituent la cible initiale. Le kit a aussi été testé de façon informelle par des collègues et les membres du laboratoire TECFA et les commentaires étaient également très positives.

D'un point de vue conceptuel et théorique, ce kit appartient à la catégorie d'objet d'apprentissage défini par la "manipulation conceptuelle" de Maria Montessori où il s'agit de manipuler de manière concrète des concepts abstraits que sont les variables et procédures dans la programmation informatique. De plus, l'outil proposé dans le cadre de ce travail permet d'avoir une approche ludique et simplifiée de l'informatique - univers qui peut "faire peur" aux novices - en permettant d'appréhender assez rapidement les bases de la programmation.

Il s'agit d'un outil non-informatique ayant pour objectif d'enseigner l'informatique, peu coûteux financièrement, adaptable et utilisable en salle de classe lors d'activités dites "débranchées". En effet, un des intérêts principaux de ce type d'outil est qu'il est éligible à une utilisation dans toutes les classes dans le cadre d'une initiation à la programmation. En effet, la dotation en équipements informatiques dans les écoles est en cours et nombre d'entre elles ne disposent pas encore des ressources informatiques permettant d'enseigner la programmation avec des logiciels comme Scratch [4] [7]. Un dernier atout majeur de ce travail est d'avoir créé un design pédagogique qui être diffusé afin d'être imprimés et/ou adaptés localement ou dans un maker space ou un fab lab.

8. Documentation, licences et fichiers SVG

Cette œuvre (design et documentation) est mise à disposition selon les termes d'une licence *Creative Commons CC BY-NC 4.0*. [8] Les dessins des différents éléments du kit ont été en partie réalisés par nos soins avec le logiciel Inkscape et en partie importés de banques d'images libres de droits. Dans ce dernier cas, les sources des images figurent dans les fichiers SVG de chaque dessin. L'ensemble des fichiers (SVG et manuel) est accessible ici

[https://edutechwiki.unige.ch/fr/STIC:STIC_III_\(2016\)/Programming_Boty](https://edutechwiki.unige.ch/fr/STIC:STIC_III_(2016)/Programming_Boty)

10. Références et liens

[1] <http://lightbot.com/>

[2] <http://www.newgrounds.com/portal/view/200730>

[3] Montessori, M. Pédagogie scientifique, la découverte de l'enfant, Desclée de Brouwer (1958)

[4] Fincher, S and I. Utting. Machines for Thinking. *Trans. Comput. Educ.* 10, 4, Article 13 (2010). DOI=<http://dx.doi.org/10.1145/1868358.1868360>

- [5] Guzdial, M., 2004. Programming environments for novices. *Computer science education research*, 2004, pp.127-154. Repéré à <http://coweb.cc.gatech.edu/mediaComp-plan/uploads/37/novice-envs2.pdf> (Avril 2017).
- [6] <http://csunplugged.org/>
- [7] Bell, Tim, Jason Alexander, Isaac Freeman, and Mick Grimley. "Computer science unplugged: School students doing real computing without computers." *The New Zealand Journal of Applied Computing and Information Technology* 13, no. 1 (2009): 20-29.
- [8] Licence de Programming Boty: <https://creativecommons.org/licenses/by-nc/4.0/>
- [9] Papert, S. *Mindstorm: Children, Computers and Powerful Ideas*, New York: Basic Books. (1980).
- [10] Brooke, J. « SUS: a « quick and dirty » usability scale ». In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland. *Usability Evaluation in Industry*. London: Taylor and Francis. (1996)
- [11] Zuckerman, O. Historical overview and classification of traditional and digital learning objects. (2006). Repéré à <https://ilk.media.mit.edu/courses/readings/classification-learning-objects.pdf>, (novembre 2016)
- [12] Le monde, 02.11.2016, En 2016, le code informatique arrive à l'école. Repéré à http://www.lemonde.fr/sciences/article/2016/11/02/en-2016-le-code-informatique-arrive-a-l-ecole_5024344_1650684.html (Avril 2017).