HAL open science

Une approche méta-design du jeu sérieux pour l'enseignement de l'informatique à l'école élémentaire

Xavier Nédélec, Bertrand Marne, Mathieu Muratet, Karim Sehaba, Jean Lapostolle

To cite this version:

Xavier Nédélec, Bertrand Marne, Mathieu Muratet, Karim Sehaba, Jean Lapostolle. Une approche méta-design du jeu sérieux pour l'enseignement de l'informatique à l'école élémentaire. Atelier " Apprendre la Pensée Informatique de la Maternelle à l'Université ", dans le cadre de la conférence Environnements Informatiques pour l'Apprentissage Humain (EIAH), Jun 2021, Fribourg, Suisse. pp.96-106. hal-03241691

HAL Id: hal-03241691

https://hal.science/hal-03241691

Submitted on 28 May 2021

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.

Une approche méta-design du jeu sérieux pour l'enseignement de l'informatique à l'école élémentaire

Xavier Nédélec[1], Bertrand Marne[2], Mathieu Muratet[3, 4], Karim Sehaba[5], Jean Lapostolle[6]
(1) ICAR UMR 5191, Université Lumière Lyon 2, France
(2) ICAR UMR 5191, Université Lumière Lyon 2, France
(3) Sorbonne Université, CNRS, LIP6, F-75005 Paris, France
(4) INS HEA, 92150 Suresnes, France
(5) Université de Lyon. Université de Lyon 2. LIRIS, CNRS, UMR5205
(6) ICAR UMR 5191, Université Lumière Lyon 2
xavier.nedelec08@gmail.com, bertrand.marne@ens-lyon.fr,
mathieu.muratet@lip6.fr, karim.sehaba@liris.cnrs.fr, lapostolle.jean@gmail.com

RESUME

Le méta-design est une approche de conception participative dans laquelle des utilisateurs finaux sont rendus concepteurs des ressources, y compris durant les phases d'utilisation. C'est à partir de cette méthode que nous souhaitons développer notre projet autour du jeu sérieux Blockly Maze visant à permettre aux enseignants du primaire d'enseigner la pensée informatique en classe. Blockly Maze est un jeu de programmation par blocs centré sur l'apprentissage de l'algorithmique. Nous voulons que les enseignants puissent l'adapter en fonction de leurs besoins d'enseignement, des parcours d'apprentissage des élèves et des usages observés sur le terrain. À cette fin, nous nous inscrivons dans une optique de recherche collaborative orientée par la conception dans laquelle des enseignantes de cycle 3 sont parties prenantes du projet. Elles contribuent à part entière au processus d'adaptation du jeu sérieux Blockly Maze et à l'élaboration de méthodes de méta-design qui se veulent génériques.

ABSTRACT

A meta-design approach to learning games for teaching computer science in primary school. Meta-Design is a participatory design approach in which end users are enabled to design an artifact, even during the use stage. It is from this method that we wish to develop our project around the serious game Blockly Maze aiming to allow primary school teachers to teach computational thinking in class. Blockly Maze is a block programming game focused on learning algorithms. We want teachers to use it to teach computational thinking and therefore be able to adapt it to their teaching needs, to the learning paths of their students and to uses observed in the field. To this end, we are using a Design Based Research approach in which teachers in primary school are involved in the project. We believe they contribute adapting Blockly Maze to their needs and to provide methods for meta-design that are intended to be generic.

MOTS-CLES : recherche collaborative orientée par la conception, genèse instrumentale, praxéologie, méta-design, jeux sérieux, pensée informatique.
KEYWORDS: design based research, instrumental genesis, praxeology, meta-design, serious games, computational thinking.

 Attribution 4.0 International.

1 Introduction

Meta-DeCT (*Meta-Design for Computer Science Teaching*) est un projet de recherche qui a pour objectif de favoriser la participation active des enseignants dans la conception et l'élaboration de jeux sérieux dans une optique d'enseignement de la pensée informatique. La méthode mise en œuvre pour élaborer ce projet s'appuie sur le méta-design (Fischer et al. 2004), une méthode intégrant l'utilisateur final dans les phases de conception et d'utilisation du jeu. Inscrit dans une démarche de recherche collaborative orientée par la conception (Sanchez et Monod-Ansaldi 2015), ce travail a pour objectif d'être un support d'aide à la conception pour les professeurs des écoles dans le but d'enseigner la pensée informatique à des élèves de cycle 3. Nous explorons plus particulièrement la contribution d'enseignants dans la (re)conception d'un jeu sérieux sur la pensée informatique. En étudiant comment intégrer des professeurs des écoles au processus de (re)conception.

Avant de développer notre approche, nous présenterons dans un premier temps notre positionnement face à certains travaux scientifiques sur la pensée informatique objet d'enseignement, sur les jeux sérieux d'apprentissage, ainsi que sur la genèse instrumentale et le méta-design. Dans un deuxième temps, nous présentons les choix méthodologiques de cette étude. Enfin, nous présenterons les premiers résultats de notre recherche sur le terrain.

2 Positionnement scientifique

2.1 La pensée informatique

De la programmation en LOGO impulsée dans les années 80 à la maîtrise d'un socle de compétences né dans les années 2000, décliné en B2i puis en C2i et dans lequel l'informatique est envisagée comme un produit utilitaire, les autorités publiques prennent peu à peu conscience des enjeux pour l'école (Baron et Drot-Delange 2016). Ainsi, la question d'un enseignement de l'informatique davantage disciplinaire a mis au jour la notion de pensée informatique. À présent, il est devenu nécessaire de comprendre ce qui se passe derrière l'écran de la machine (Baron et Bruillard 2008). Ainsi, les approches didactiques se développent et se diversifient avec Scratch (Resnick et al. 2009), l'informatique débranchée (Bell, Witten, et Fellows 1998) ou encore la robotique pédagogique (Spach 2017).

Qu'entend-on précisément par « *pensée informatique* » ? Jeannette Wing le rappelle (Wing 2006) « *la pensée informatique est un ensemble d'attitudes et d'acquis universellement applicables que tous, et pas seulement les informaticiens, devraient apprendre et maîtriser* » [1].

La réflexion de Wing a été étayée et certains ont décliné la pensée informatique en une série de compétences structurant le domaine en cinq parties qui permettant de résoudre un problème : (1) pensée algorithmique, (2) abstraction, (3) évaluation, (4) décomposition, (5) généralisation (Brunet et al. 2020).

La question de la pensée informatique fait toujours débat quant à une définition précise et acceptable pour pouvoir l'enseigner. Gilles Dowek estime qu'il faut nécessairement transmettre l'image d'une discipline la plus fidèle possible et propose ainsi de décliner l'informatique en quatre concepts : algorithme, langage, machine et information (Dowek 2011). Alors que d'autres considèrent

[1] *It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.*

qu'enseigner l'informatique avec le moteur Scratch permet de développer une pensée algorithmique (Tchounikine 2017).

En nous fondant sur tous ces travaux, nous proposons une description de la pensée informatique qui sera mise en œuvre dans notre étude. Nous pensons qu'il s'agit tout d'abord d'identifier un problème en le reformulant de façon simple et claire. Vient ensuite sa décomposition. Cela consiste à organiser et analyser les données : à savoir décomposer le problème en sous-problèmes pour proposer une amorce de solution par étapes. Cette phase de décomposition suggère également une phase d'abstraction du problème afin d'identifier un système de représentation dont le but est d'élaborer une modélisation du système. Cette modélisation s'appuie ensuite sur le choix des représentations des éléments du système, des modélisations et du choix des interactions. Une fois le problème abstrait, il s'agit d'exprimer une solution algorithmique en langage naturel puis de la traduire en langage machine. L'itération de la solution permet d'analyser les réussites et échecs. Il convient à cet égard d'identifier les problèmes rencontrés, de comparer la proposition à d'autres solutions. Le test est une épreuve qui ouvre ensuite sur la généralisation du problème à d'autres situations à savoir : (1) identifier les conceptions sous-jacentes au nouveau problème ; (2) reconnaître des cas identiques ou légèrement différents ; (3) généraliser l'algorithme pour des cas similaires.

Avec cette vision de la pensée informatique, notre ambition est de permettre son apprentissage grâce à un jeu sérieux exploitable par des enseignants non experts du domaine. Notre problématique est de développer des instruments adaptés et adaptables à leurs capacités et à leurs besoins.

Dans la sous-section suivante, nous inscrivons nos travaux dans l'état de l'art sur les jeux sérieux d'apprentissage.

2.2 Les jeux sérieux d'apprentissage

Le jeu sérieux d'apprentissage est un outil qui combine une activité hédonique avec un message « sérieux » (Djaouti 2011). Par conséquent, il peut être un élément motivationnel pour l'apprenant, car il intègre l'apprentissage dans un cadre ludique (Garris, Ahlers, et Driskell 2002).

Pour notre projet nous nous appuyons sur le jeu Blockly Maze (voir FIGURE 1), un Logiciel Libre développé par Google avec un code source disponible et réutilisable, sensiblement inspiré de Scratch (Resnick et al. 2009). Les ressorts et les mécaniques du jeu Blockly Maze visent un apprentissage de la programmation, de l'algorithmique et du déplacement dans l'espace par la résolution de problèmes. Blockly Maze est considéré comme un jeu sérieux d'apprentissage, car les problèmes à résoudre sont posés par le biais d'une interface ludique : il s'agit pour le joueur de faire sortir un avatar d'un labyrinthe grâce à de la programmation par blocs. Le jeu permet des essais multiples. Avec une interface simple à prendre en main, où les informations visuelles ne sont pas abondantes, le joueur est rapidement concentré sur la tâche à résoudre. Le joueur est mis au défi dans une optique motivationnelle où la progression se fait à travers dix niveaux, avec des contraintes progressives, par exemple, un nombre restreint de blocs à utiliser.

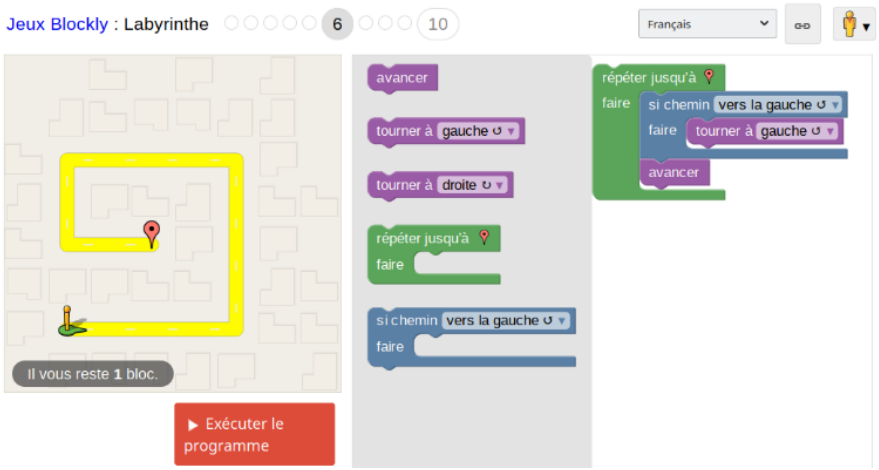

FIGURE 1 : capture d'écran du niveau 6 de Blockly Maze avec le bloc de programmation du parcours de l'avatar

Cependant, ces 10 niveaux nous paraissent trop restreints au regard des possibilités de travail sur la pensée informatique. Nous estimons qu'il faut diversifier l'offre de contenu, de parcours à travailler afin d'englober le plus largement possible la notion de pensée informatique et donc étoffer le nombre de niveaux. Nous estimons également que si nous voulons que l'enseignant soit partie prenante de la conception, il faut lui donner la possibilité de choisir, modifier et orchestrer ses niveaux dans des scénarios adaptés à ses objectifs pédagogiques et au niveau de l'élève. De plus, l'intégration d'un jeu sérieux d'apprentissage comme Blockly Maze pose des questions d'appropriation par les enseignants (Sardone 2018).

Ainsi, pour favoriser une large adoption et une bonne utilisation de Blockly Maze, nous fondons notre travail sur les concepts de genèse instrumentale et de méta-design.

2.3 Genèse instrumentale et méta-design

La genèse instrumentale génère un double processus de conception : l'instrumentalisation orientée vers l'artefact où l'utilisateur confère à l'objet de nouvelles attributions pour répondre à ses besoins, puis l'instrumentation centrée sur le sujet (l'utilisateur) où ce dernier utilise « *des schèmes* » déjà existants pour en comprendre l'usage (Rabardel 1995).

Le méta-design c'est « *concevoir le processus de conception* » (Fischer et al. 2004) : ce processus permet aux utilisateurs de résoudre eux-mêmes des problèmes rencontrés en cours d'utilisation d'un artefact en leur offrant la possibilité d'être impliqués dans le processus de conception initiale et dans une démarche de conception dans l'usage. Cette démarche de conception dans l'usage peut être suscitée grâce à l'*« underdesign »* (Fischer et al. 2004) : il s'agit de proposer des artefacts très modulaires non finalisés laissant la possibilité aux utilisateurs de les adapter selon leurs besoins et leurs usages. L'approche par le méta-design est donc une méthode d'instrumentation, voire d'instrumentalisation, elle est par ailleurs est étudiée et mise en œuvre dans le contexte des jeux d'apprentissage (Marne 2014; Vermeulen 2018).

Par conséquent, notre méthodologie de recherche s'inscrit dans une réciprocité entre le chercheur et l'enseignant-utilisateur où chacun se nourrit des apports de l'autre au cours des travaux de conception menés sur Blockly Maze.

3 Méthodologie de recherche

Nos travaux de recherche s'inscrivent dans une démarche de Recherche Collaborative Orientée par la Conception (ROC) (Sanchez et Monod-Ansaldi 2015). Notre but, est de favoriser l'utilisation d'un jeu sérieux d'apprentissage comme Blockly Maze par des enseignants du primaire pour enseigner la pensée informatique par une démarche de méta-design fondée sur l'underdesign.

Dans notre démarche de ROC, nous avons recours à un *« broker »* (Sanchez et Monod-Ansaldi 2015), c'est-à-dire une personne jouant un rôle d'intermédiaire entre les chercheurs (et également experts sur la pensée informatique et les jeux sérieux) et les enseignants (plus ou moins expérimentés).

Avec l'aide du « *broker* » et afin d'ancrer notre démarche de méta-design, nous envisageons un travail collaboratif en trois grandes phases. Dans un premier temps, nous recrutons des enseignants volontaires, puis en les questionnant sur leur parcours professionnel, sur leur expérience de l'informatique, et sur leur enseignement de la discipline en classe nous dressons un portrait de leur expérience professionnelle ainsi que de leur rapport à l'informatique et à son enseignement. Dans un second temps, nous nous servons de la construction d'une carte heuristique de la pensée informatique avec les enseignants pour explorer et comprendre quels sont leurs préconceptions, leurs usages et leurs besoins dans l'enseignement de ce domaine. Enfin, dans un dernier temps, nous élaborerons avec eux les modifications à apporter à Blockly Maze pour qu'il corresponde à leurs besoins pédagogiques et qu'il puisse être utilisé, et adapté par d'autres enseignants.

Lors de la seconde phase, inscrite dans le cadre d'une ROC, le recours à un *« broker »* permet de tenir un discours commun entre chaque partie : universitaires et enseignants. Concrètement, la praxéologie que nous souhaitons mettre en œuvre prend la forme de la conception d'une carte heuristique sur la pensée informatique élaborée avec les enseignants. Ici, le *« broker »* joue un rôle d'intercesseur entre les deux parties en aidant notamment les enseignants *« à accoucher »* leurs idées, leurs propositions dans le cadre défini par notre étude.

Cependant, la carte heuristique[2] n'est principalement qu'un outil de discussion, de réflexion de la notion de pensée informatique construite conjointement. L'objectif n'est aucunement de définir ce que doit être la pensée informatique, mais plutôt de servir d'élément discursif, nécessaire à l'étayage du concept et également d'être un support de conception et d'intercession entre le chercheur et le praticien. Cette carte sera en outre un support de formation pour les enseignants novices sur les questions de pensée informatique, un support d'indexation de compétences qui figureront dans les outils auteurs.

Dans la troisième phase de la recherche, nous allons utiliser un outil auteur de scénarios adapté au méta-design : APPLiq (Marne 2014). Les enseignants ont la capacité d'y agir directement sur le jeu en proposant un parcours de niveaux à l'apprenant. À ces niveaux sont indexés des prérequis en entrée et des objectifs en sortie, de sorte que l'enseignant choisisse quelles compétences l'apprenant va travailler.

L'agilité du dispositif rend possible de changer de niveau en fonction des performances du joueur. Il n'est pas nécessaire de suivre la progression dans le jeu de manière linéaire. Par exemple, si

[2]Voici un exemple de carte heuristique produite avec des enseignantes, selon les mêmes principes, dans une étude préliminaire de celle présentée ici : https://mycore.core-cloud.net/index.php/s/F4FyN9tbJJxOkKq/download (vu le 21/05/2021)

l'enseignant estime que l'apprenant remplit les prérequis et les objectifs d'un niveau, il peut lui attribuer de jouer sur un nouveau niveau. L'enseignant décide librement de suivre ou de modifier le parcours de chaque apprenant. APPLiq indique aussi à l'enseignant-utilisateur si des incohérences pédagogiques apparaissent au cours de la séquence de jeu (icônes triangulaires jaunes sur les liens entre les niveaux présentés dans la FIGURE 2). Ces incohérences sont liées à des prérequis n'entrant pas dans les objectifs des niveaux précédemment joués. Toutefois, pour respecter sa pleine liberté pédagogique, l'enseignant peut passer outre ce signal et confirmer son choix.

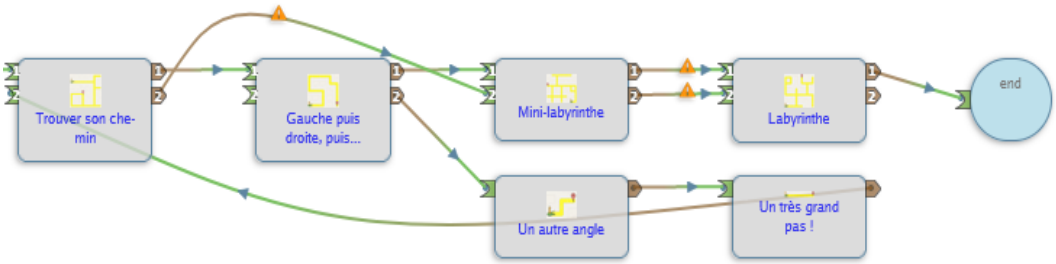

FIGURE 2 : capture d'écran d'un exemple de scénario de Blockly Maze modélisé dans APPLiq

Notre ambition est évidemment d'offrir la possibilité aux enseignants de prendre en main APPLiq afin qu'ils puissent choisir par eux-mêmes les niveaux du jeu travaillés par les apprenants. Même si son utilisation demeure assez simple, il faut néanmoins qu'il soit compris des utilisateurs non experts, comme pourraient l'être certains enseignants. Le recours à un *« broker »* jouant un rôle d'intermédiaire entre l'expert et le non-expert vise à favoriser la réussite du projet.

Toutefois, si APPLiq permet la construction de progressions non-linéaires dans le jeu, il ne permet pas de modifier l'organisation interne de chaque niveau. Pour cela, l'enseignant devrait donc intervenir directement sur le code source du jeu. C'est pourquoi, dans le contexte de notre démarche de méta-design, nous souhaitons aussi élaborer avec les enseignants volontaires un outil auteur de niveaux, complémentaire d'APPLiq, permettant de faciliter ces modifications.

Dans la section suivante, nous proposons les premiers résultats de nos travaux concernant la première phase (les entretiens de profilage et la prise en main de Blockly Maze) et le début de la seconde phase (la construction de la carte heuristiques).

4 Premiers travaux sur le terrain

4.1 Phase 1 : Recrutement et profilage des volontaires

C'est le « *broker* », lui-même enseignant de CM2, formé à l'ingénierie pédagogique dans le contexte des EIAH, qui a recruté les enseignantes volontaires de notre étude. Elles enseignent en cycle 3, l'une en CM1, les deux autres en CM2, toutes en classe ordinaire. Elles ont accepté de participer au projet, car elles y voient une possibilité de formation continue et une montée en compétences. Nous présentons plus loin des extraits de leurs entretiens au sujet de leur perception de l'informatique, de leur usage en classe et de leur retour à la suite de la prise en main du jeu Blockly Maze.

Dans un premier temps, afin de dresser un tableau descriptif du profil de chaque volontaire, nous avons mené un entretien individuel en évitant d'interférer ou biaiser certaines de leurs réponses. Nous les avons interrogées sur leur parcours professionnel, leur rapport personnel puis professionnel aux outils numériques et à l'informatique et sur leur enseignement de l'informatique.

Dans un second temps, les enseignantes ont pris en main le jeu Blockly Maze, sans information au préalable. À l'issue de cette partie qui a été enregistrée (audio et écran), un débriefing à chaud nous a permis de sonder leurs impressions et jugements à propos de Blockly Maze. Puis, quelques jours plus tard, lors d'un second débriefing, nous les avons questionnées plus en profondeur sur ce qu'elles avaient appris du jeu et sur les dispositifs didactiques qu'elles utiliseraient en classe.

La première des participantes impliquées dans le projet enseigne en CM1. Elle a 40 ans et a 16 ans d'ancienneté dans le métier. Même si elle a voulu au départ devenir professeure de portugais, elle a passé le concours de professeur des écoles et a toujours connu la classe. C'est une enseignante expérimentée qui connaît très bien le CM1 puisque cela fait dix ans qu'elle enseigne à ce niveau.

Les deux autres enseignantes sont quant à elles moins expérimentées. L'une, âgée de 52 ans, a trois ans d'expérience, l'autre, âgée de 43 ans, a sept ans d'expérience. Toutes deux découvrent le CM2. Elles ont aussi en commun d'avoir un parcours professionnel diversifié, notamment hors du professorat. L'un des points communs que nous identifions chez ces trois enseignantes concerne la perception et l'appréhension qu'elles ont de l'informatique. La vision qu'elles en ont est confuse et elles se sentent dépassées par l'objet « *ordinateur* » : « *Je me sens vieille un peu et c'est un outil avec lequel je ne suis pas très aguerrie* », *« je me sens perdue* », « *on va dire que mon fils de 13 ans en connaît beaucoup plus que moi et je fais souvent appel à lui* ».

Elles posent un regard distant et perçoivent l'informatique exclusivement comme un outil de travail professionnel destiné à la pratique de la bureautique pour préparer leurs cours et faire des recherches sur internet. L'école dans laquelle elles travaillent est équipée de TNI et tablettes. Elles utilisent le TNI pour la projection des cours, mais ne se servent pas du logiciel accompagnant son usage. Concernant l'enseignement de l'informatique en classe, seule l'enseignante de CM1 estime avoir déjà enseigné la discipline à ses élèves, mais dans une démarche d'informatique exclusivement utilitaire, comme se servir d'un traitement de texte.

Leur témoignage soulève une demande forte de formation à l'outil informatique : « *(...) c'est vrai que l'on n'a pas vraiment eu de formation sur l'informatique. C'est difficile, il faut chercher par nous-même* », « *j'ai envie d'être formée, de découvrir pour pouvoir enseigner l'informatique plus tard, de prendre les choses en main* ».

Seule l'enseignante de CM1 a reçu une formation en robotique « (…) *on nous a montré un robot-souris qui se déplaçait sur un tapis et qui avait été programmé par des élèves au préalable* ». Mais l'équipement est inexistant et les concepts (programmation, fonctionnement d'une machine) sous-jacents à cette formation n'ont pas été évoqués.

Nous pouvons alors nous interroger sur la formation : quel genre de formation souhaitent-elles suivre ? Une formation à l'usage du numérique ? Ou bien une formation destinée à l'apprentissage de l'informatique ? Les réponses recueillies montrent que cette distinction reste floue dans leur esprit.

À propos de la question sur la pensée informatique, les trois enseignantes disent ne jamais avoir entendu parler de cette notion. Deux d'entre elles connaissent Scratch, découvert à l'occasion du suivi de travail de leurs enfants au collège, mais nullement dans le cadre de leur métier. Enfin, elles énoncent clairement la difficulté de se projeter dans un enseignement de l'informatique, ce qui est évidemment compréhensible compte tenu de leur niveau de connaissance du domaine.

En respectant notre postulat de départ d'intégrer l'utilisateur à la conception de notre projet, la première étape a été de mettre entre les mains des trois enseignantes le jeu Blockly Maze. Le but était double : (1) Observer la manipulation du jeu et leur progression ; (2) Recueillir leurs impressions sur les dimensions ludiques et pédagogiques.

Pour les trois enseignantes, dès le niveau 1 les objectifs ludiques du jeu n'étaient pas identifiés : la manipulation des blocs ainsi que le glisser-déposer dans l'espace réservé à l'écriture du programme. Il fallait un guidage oral pour débloquer la situation. Il a fallu deux minutes pour réussir le niveau 1 pour deux enseignantes et plus de quatre minutes pour la troisième. Par la suite, pour deux d'entre elles, la progression s'est faite sans trop de difficultés. Il ne leur a fallu qu'entre quatre et cinq minutes pour réaliser les niveaux 2, 3, 4, 5. En revanche pour une enseignante de CM2 la progression a été plus compliquée. Quinze minutes ont été nécessaires pour réussir les niveaux 2, 3, 4, 5. Enfin, pour toutes les trois, le niveau 6 n'a pas été réussi. Nous avons arrêté au bout d'un quart d'heure d'essai.

À propos des objectifs pédagogiques, elles ont compris qu'il fallait faire cheminer l'avatar vers une cible (sortie du labyrinthe).

Nous en concluons, que la progression dans le jeu a été hésitante pour chacune d'elle. Programmer une séquence d'instructions a été correctement exécuté. L'utilisation de la boucle « *repeat until* » a nécessité de nombreux essais sur les niveaux 3, 4, 5, mais au niveau 6, aucune n'a su imbriquer ce bloc dans un autre bloc du programme. D'ailleurs, c'est aussi à ce niveau qu'elles ont toutes les trois été bloquées à cause notamment de l'apparition de la condition « *if/else* » qu'elles n'ont pas su employer.

Le jeu met évidemment en exergue des compétences de résolution de problèmes liées à l'algorithmique. À partir de leurs essais-erreurs, de leurs interrogations sur leurs échecs, de leurs réussites, nous avons observé leur démarche de raisonnement et les procédures employées pour résoudre les problèmes de chaque niveau. Même si la programmation par bloc leur est inconnue, le raisonnement adopté pour résoudre un niveau est souvent similaire : décomposer le cheminement de l'avatar, planifier les étapes du trajet, associer instructions et/ou imbrications d'instructions dans une boucle. Cependant, lors de leur propre retour d'expérience, aucune n'a su faire le lien entre le jeu et l'apprentissage de l'informatique ni perçu des contenus sous-jacents de la pensée informatique de chaque niveau.

En d'autres termes, personne n'a su dire quelles compétences d'informatique étaient travaillées. Nous expliquons ce constat tout simplement par le fait qu'aucune n'a conscience de posséder de connaissances ou compétences en informatique. Néanmoins, le point qui a soulevé leur attention, et que nous n'avions pas envisagé, concerne la question de la représentation et de l'orientation dans l'espace du plan du jeu et l'orientation d'un objet par rapport à un autre. L'enseignante de CM1 a relié cette caractéristique à un apprentissage en cours dans les niveaux de CM1 et CM2 qui s'inscrit en géométrie. Cette remarque nous semble importante et nous devons la prendre en compte pour proposer un artefact qui doit lever toutes formes d'obstacle à l'apprenant-joueur.

Enfin, même si elles n'ont pas encore conscience des concepts de programmation que permet d'apprendre Blockly Maze, les trois enseignantes s'accordent à dire que ce jeu peut être un bon outil à utiliser en classe. Elles estiment que proposer un tel objet constituerait un nouveau support ludique, source de motivation.

4.2 Phase 2 : ébauche de construction d'une carte heuristique

Pour la seconde phase du projet, celle de la construction de la carte, nous avons opté pour une démarche praxéologique à savoir la constitution d'un lexique pouvant définir des concepts de la pensée informatique et de leurs liens.

Nous souhaitons mettre en œuvre un nouveau protocole d'observation (avec enregistrement audio et vidéo) des enseignantes jouant à Blockly Maze. Mais, cette fois-ci, il leur sera demandé de vocaliser chacune de leurs actions, chacun de leurs raisonnements au cours de la partie, de sorte que l'on puisse dresser un vocabulaire et un raisonnement type sur lequel nous pourrions nous appuyer pour étayer la carte heuristique.

Nous n'avons pas encore réalisé cette expérience en leur présence. Toutefois, nous nous sommes essayés à la pratique en vocalisant nos parties. Deux observations peuvent être soulignées : tout d'abord, l'exercice de *« penser à voix haute »* n'est pas aussi aisé qu'il parait. Ensuite, le lexique basique de programmation est présent (instruction, boucle, *repeat*/*until*, les conditions) et les raisonnements sont clairement identifiés : la question de l'anticipation, la décomposition, l'essai-erreur, le test de la solution, la généralisation d'une solution. Si de notre côté, les résultats sont ceux auxquels nous nous attendons, qu'en sera-t-il des enseignantes ? Probablement, que le lexique spécifique de la programmation ne sera pas prononcé, mais des mots s'en apparentant pourront émerger. D'autre part, la façon de raisonner ne sera peut-être pas aussi claire, mais nous pourrons observer des points de convergence. Dans tous les cas, ce premier travail constitue pour elles un début de formation à la pensée informatique, car il sera suivi d'un retour réflexif sur leur méthode de résolution et d'un apport de vocabulaire spécifique.

Ce premier travail doit nous amener vers la production d'un outil commun de pensée, de réflexion et de construction du concept de pensée informatique. Puis, dans un second temps, il doit permettre la constitution d'un socle pour co-construire à partir de l'expérience des enseignantes et de leur connaissance des compétences des élèves des niveaux de jeu plus adaptés.

5 Conclusion

Le méta-design ancre l'idée d'engager activement les utilisateurs-enseignants dans la conception de jeux sérieux afin que ces derniers puissent les intégrer dans leurs parcours pédagogiques et favoriser ainsi l'apprentissage des élèves. Faire participer des enseignants *« novices »* en informatique à un projet de conception d'un jeu sérieux nécessite d'emblée de planifier un accompagnement formatif. Le recours à une personne jouant le rôle de *« broker »* nous permet d'avoir un retour pragmatique des enseignantes sur les conditions d'usage d'un tel dispositif pour elles-mêmes et en classe, en situation d'enseignement avec des élèves. Les enseignantes nous apportent quant à elles leur expertise de pédagogue sur les connaissances et compétences des élèves de cycle 3 afin d'adapter au mieux Blockly Maze.

Notre démarche de recherche collaborative orientée par la conception vise à prendre en compte intégralement les compétences et connaissances des enseignantes en matière d'informatique, sans contournement, tel qu'elles se présentent en condition écologique. Ce choix nous impose de surcroit un accompagnement de proximité pour outiller les enseignantes afin d'optimiser leur participation et leur adhésion au projet. Nous avons conscience du problème de connaissance du domaine de la pensée informatique, propre d'ailleurs à beaucoup d'enseignants du primaire, en raison d'un manque de formation. Nous estimons que l'apport d'un « *broker* », servant de pont entre la recherche et le terrain,

ne peut que bénéficier aussi bien au projet qu'aux enseignantes elles-mêmes. Et l'apport d'une carte heuristique sur la pensée informatique est justifié d'une part par le besoin en formation du domaine, d'autre part par celui d'établir un socle commun sur lequel construire la démarche de méta-design.

Fondés sur ces premiers pas, nos travaux en cours et futurs portent sur le développement de nouveaux niveaux de Blockly Maze et leur intégration dans APPLiq pour la scénarisation ainsi que sur la mise en place des indicateurs de suivi pour permettre aux enseignants de comprendre les usages de leurs élèves. Toutes ces actions se font en collaboration avec les enseignantes sur le terrain avec qui nous continuons les entretiens et expérimentations.

Références

Baron, Georges-Louis, et Eric Bruillard. 2008. « Technologies de l'information et de la communication et indigènes numériques : quelle situation ? » *STICEF* 15 (195): 12 pages.

Baron, Georges-Louis, et Béatrice Drot-Delange. 2016. « L'informatique comme objet d'enseignement à l'école primaire française? Mise en perspective historique ». *Revue française de pédagogie. Recherches en éducation*, nº 195: 51-62.

Bell, Tim, Ian H. Witten, et Mike Fellows. 1998. *Computer Science Unplugged: Off-line activities and games for all ages*. Citeseer.

Brunet, Olivier, Amel Yessad, Mathieu Muratet, et Thibault Carron. 2020. « Vers un modèle de scénarisation pour l'enseignement de la pensée informatique à l'école primaire ». In *Didapro 8 – DidaSTIC*. Lille, France. https://hal.archives-ouvertes.fr/hal-02496191.

Djaouti, Damien. 2011. « Serious Game Design : considérations théoriques et techniques sur la création de jeux vidéo à vocation utilitaire ». http://thesesups.ups-tlse.fr/1458/.

Dowek, Gilles. 2011. « Les quatre concepts de l'informatique ». In , 21-29. Athènes : New Technologies Editions. https://edutice.archives-ouvertes.fr/edutice-00676169.

Fischer, Gerhard, Elisa Giaccardi, Yunwen Ye, Alistair G. Sutcliffe, et Nikolay Mehandjiev. 2004. « Meta-design: a manifesto for end-user development ». *Communications of the ACM* 47 (9): 33-37.

Garris, Rosemary, Robert Ahlers, et James E. Driskell. 2002. « Games, motivation, and learning: A research and practice model ». *Simulation & gaming* 33 (4): 441-67.

Marne, Bertrand. 2014. « Modèles et outils pour la conception de jeux sérieux : une approche meta-design ». Thèse de Doctorat en Informatique, Paris: Université Pierre et Marie Curie (UPMC).

Rabardel, Pierre. 1995. *Les hommes et les technologies : une approche cognitive des instruments contemporains*. U. Série Psychologie. Paris, France: Armand Colin. http://ergoserv.psy.univ-paris8.fr/Site/Groupes/Modele/Articles/Public/ART372105503765426783.PDF.

Resnick, Mitchel, John Maloney, Andrés Monroy-Hernández, Natalie Rusk, Eastmond Eastmond, Karen Brennan, Amon Millner, et al. 2009. « Scratch: Programming for All ». *Communications of the ACM* 52 (11): 60-67. https://doi.org/10.1145/1592761.1592779.

Sanchez, Éric, et Réjane Monod-Ansaldi. 2015. « Recherche collaborative orientée par la conception. Un paradigme méthodologique pour prendre en compte la complexité des situations d'enseignement-apprentissage ». *Éducation et didactique*, nº 9-2 (septembre): 73-94. https://doi.org/10.4000/educationdidactique.2288.

Sardone, Nancy B. 2018. « Attitudes Toward Game Adoption: Preservice Teachers Consider Game-Based Teaching and Learning ». *International Journal of Game-Based Learning (IJGBL)* 8 (3): 1-14. https://doi.org/10.4018/IJGBL.2018070101.

Spach, Michel. 2017. « Activités robotiques à l'école primaire et apprentissage de concepts informatiques: quelle place du scénario pédagogique? Les limites du co-apprentissage ». Thèse de doctorat de Sciences de l'éducation, Université Sorbonne Paris Cité.

Tchounikine, Pierre. 2017. « Initier les élèves à la pensée informatique et à la programmation avec Scratch ».

Vermeulen, Mathieu. 2018. « Une approche meta-design des learning games pour développer leur usage ». Theses, Sorbonne Université, Faculté des Sciences et Ingénierie. https://tel.archives-ouvertes.fr/tel-01871048.

Wing, Jeannette M. 2006. « Computational Thinking ». *Communications of the ACM* 49 (3): 33-35.