

Co-cr ation d'une s quence d'introduction   l'informatique avec des enseignants non-informaticiens : retour d'exp rience

Olivier Goletti
Universit  Catholique de Louvain
prenom.nom@uclouvain.be

R sum . Dans le contexte limit  de l'enseignement de l'informatique dans le syst me  ducatif obligatoire en Belgique, cet article pr sente le processus de mise en place d'une s quence de cours co-cr e par l'auteur et trois enseignants de la deuxi me ann e du secondaire inf rieur. Le processus d' laboration de cette s quence ainsi que les activit s, test es dans trois classes, sont pr sent s. Une  valuation sommative a  t  r alis e pour estimer les acquis des  l ves au terme de cette s quence. Celle-ci est compl t e par une analyse qualitative r alis e aupr s des  l ves et des enseignants. Cet article pr sente la co-cr ation de cette s quence et propose des pistes de r utilisation dans le cadre de la formation continue d'enseignants pour l'informatique   l' cole.

Mots-cl s. Formation d'enseignants, enseignement obligatoire, formation continue, informatique pour tous,  ducation   l'informatique.

1 Introduction

En Belgique francophone, il n'y a pas de cours d'informatique obligatoire dans le syst me  ducatif (enseignement fondamental et secondaire). Avant 14 ans, un tel cours ne peut  tre organis  par les  coles qu'  titre d'activit  compl mentaire. Chaque  cole dispose en effet de deux   quatre p riodes par semaine dans l'horaire des  l ves pour organiser diverses activit s compl mentaires au premier degr  du secondaire, l'une d'elles pouvant  tre de l'informatique. Ce cours facultatif d'informatique n'est donc accessible que dans certaines  coles. En outre, ce dernier est avant tout un cours de litt ratie num rique au sens large, voire m me de bureautique en pratique (exemple de r f rentiel pour cette activit  : FESeC (2007)).

Cette absence d'une offre de formation de tous nos jeunes   l'informatique prendra-t-elle bient t fin ? Une r forme de l'enseignement obligatoire est bien en cours, le Pacte pour un enseignement d'excellence¹, mais l'informatique n'y est pr sente qu'en filigrane, cach e dans le num rique   la fin d'un intitul    rallonge : la formation manuelle, technique, technologique et num rique². Pourtant plusieurs associations et organisations publient des appels   enseigner les fondements de l'informatique (y compris la pens e informatique et les bases de la programmation)   l' cole tant en Belgique, qu'aux niveaux europ en ou international : (Gander et al., 2013), (Caspersen, Gal-Ezer, McGettrick, & Nardelli, 2018), (SI², 2017), (Bersini et al., 2018), la sortie du l'union wallonne des entreprises³, et bien d'autres. Plusieurs initiatives p riscolaires (comme des clubs d'apprentissage de la programmation⁴), post scolaires ( cole 19⁶, Be-code⁷, etc.) voire d'animation dans le cadre scolaire (Kodo wallonies⁸, le PASS⁹, etc.) se mettent en place. Mais cela ne pourra  voluer vers une solution satisfaisante et durable que si les enseignants investissent ce champ de connaissances et comp tences. En effet, l' ducation de tous aux bases de l'informatique doit passer par l'enseignement obligatoire et donc par ses enseignants. Ceux-ci ont pour se faire besoin d'une formation comme le montre l'enqu te effectu e par (Henry & Smal, 2018).

En ce qui concerne la formation initiale des enseignants (FIE) en Belgique francophone, il n'existe   ce jour que peu de projets de formation des enseignants   l'informatique, si ce n'est le tout jeune retour de l'AESS¹⁰ informatique   l'ULB¹¹. Pourtant d'autres pays qui mettent en place des initiatives institutionnalis es

¹ <http://www.pactedexcellence.be/>, sauf indication contraire les sites ont  t  consult s le 10/05/18

² D cret mission, Article 60septies, http://www.galilex.cfwb.be/document/pdf/21557_031.pdf

³ <http://www.uwe.be/enseignement-et-formation-priorite-des-entreprises/>

⁴ <http://www.thecodingclub.be/>

⁵ <https://www.coderdojobelgium.be/fr>

⁶ <https://www.s19.be/>

⁷ <https://www.becode.org/>

⁸ <http://kodowallonie.be/>

⁹ <http://www.pass.be/je-suis/une-ecole/les-aventures-numeriques-a-l-ecole-1.htm?lng=fr>

¹⁰ AESS : agr gation de l'enseignement secondaire sup rieur permettant d'enseigner aux  l ves des 3 derni res ann es de l'enseignement obligatoire

¹¹ http://banssbfr.ulb.ac.be/PROD_frFR/bzscsrse.p_disp_prog_detail?term_in=201718&prog_in=AG-INFO&lang=FRENCH

d'enseignement de l'informatique pour tous ont mis en place de telles formations initiales pour enseignants (Gal-Ezer & Stephenson, 2014). En Belgique francophone, une r forme de la FIE est aussi dans les cartons mais rien ne semble garantir actuellement son alignement avec la mise   jour des r f rentiels envisag e par le pacte. En particulier pour l'informatique qui serait une nouvelle discipline dans les r f rentiels, on peut craindre qu'il faille du temps pour qu'elle trouve sa place au sein d'une r forme par ailleurs importante et donc d j  « difficile   g rer » de la FIE. Et au-del  de la formation initiale, il faudra imp rativement tenir compte de tous les enseignants d j  en fonction et donc assurer leur formation.

C'est pourquoi la voie privil gi e pour l'instant est la formation continue. C'est la t che   laquelle SI² s'attelle. SI² est un groupe de travail constitu  des universit s et de la plupart des hautes  coles sections informatiques francophones belges qui ont  t  mandat es par la R gion Wallonne, en accord avec la F d ration Wallonie Bruxelles, pour d velopper des offres de formation dans ce domaine. Apr s avoir publi  son propre appel   l'int gration de l'informatique   l' cole (SI², 2017) et avoir  tabli un r f rentiel des comp tences attendues en fin de tronc commun¹², ce papier pr sente l' laboration d'une s quence de cours co-cr e e avec des enseignants.

Cet article est donc un retour d'exp rience sur la cr ation de cette s quence de cours. Il a aussi pour objectif de montrer qu'il est possible pour un enseignant non expert de donner une s quence de cours sur des concepts importants de l'informatique s'il b n ficie d'une formation et d'un accompagnement. La section suivante d taillera le contexte   la gen se de ce travail et les objectifs derri re cette collaboration. Ensuite seront d taill s le processus de mise en place ainsi que la s quence elle-m me. La section 4 introduira les r sultats qualitatifs issus d'observations en classe et du retour des  l ves et enseignants. L'article se terminera par une conclusion et les suites envisag es   ce travail.

2 Contexte,  tat de l'art et objectifs

Dans le cadre de sa mission, SI² a principalement mis en avant trois raisons pour souligner l'int r t de donner des cours d'informatique dans le cursus obligatoire de tous les  l ves :

- d velopper une culture g n rale autour de l'informatique omnipr sente dans notre quotidien, incluant la compr hension des concepts fondamentaux sous-jacents, au m me titre que celle que procure le syst me scolaire dans les autres disciplines scientifiques,
- d couvrir cette discipline m connue pour permettre   tous de s'orienter ou non vers cette fili re en connaissance de cause,
- contribuer   trouver une r ponse   la p nurie structurelle de main-d' uvre qualifi e dans le secteur num rique.

Comme mentionn  en introduction, SI² a publi  une proposition de r f rentiel de comp tences en sciences informatiques, destin  au premier degr  du secondaire et ax  autour de cinq domaines fondamentaux : repr sentation des donn es, algorithmique, programmation, mat riel et enfin r seaux et s curit . Ce document a  t  r dig  en s'inspirant notamment d'initiatives comparables dans d'autres pays^{13,14}.

Sans curriculum officiel en informatique et sans formation initiale ni continue des enseignants dans cette discipline, on se trouve dans une situation peu propice pour trouver des candidats enseignants d sirieux de faire de l'informatique en classe. En effet, d'une part, ils n'y voient pas n cessairement d'int r t vu la non-pr sence dans le curriculum et d'autre part, ceux qui sont malgr  tout int ress s n'ont pas n cessairement le temps ou la formation pour r aliser ces activit s en classe. C'est un cercle vicieux qu'on observe aussi dans d'autres pays qui se sont lanc s dans des r formes similaires d'int gration de l'informatique   l' cole (Brown, Sentance, Crick, & Humphreys, 2014).

Dans la perspective d'un  ventuel passage   l' chelle si l'informatique est int gr e dans le curriculum de l'enseignement obligatoire, la question de la formation des enseignants est vraiment critique. Les modalit s de cette formation doivent donc  tre pens es en cons quence. Nous aurions pu aller tester nous-m mes une s quence cr e e sans concertation avec des enseignants mais cela n'aurait que tr s peu impliqu  les enseignants, simples observateurs de ce qui se passerait dans leur classe. Nous aurions pu fournir une activit  « cl  sur porte » aux enseignants mais ils n'auraient alors pas n cessairement adh r    la mani re dont l'activit  avait  t  cr e e. Cela aurait donc nuit   la p rennit  du projet. Nous avons donc opt  pour la co-cr ation des activit s impliquant l'enseignant d s le d but du projet. Un enseignant ainsi form  pourra continuer   donner la s quence par apr s, il

¹² https://sicarre.be/pmwiki/uploads/Documents/referentiel_sciences_info_si2.pdf

¹³ <https://k12cs.org/>

¹⁴ <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>

sera capable de l'adapter au fil des ann es et ainsi il touchera plusieurs g n rations d' l ves pendant que nous pourrions nous consacrer   la formation et l'accompagnement d'autres enseignants. La volont  de SP² de toucher tous les  l ves dans le cadre de l'enseignement obligatoire est d pendante d'une telle croissance.

Notre volont   tait donc de trouver des enseignants int ress s a priori et avec qui nous pourrions co-cr er une s quence de cours qui r pondrait   leurs attentes. Ainsi nous  tions assur s de sa coh rence avec leur pratique et nous pouvions esp rer que la s quence de cours cr ee corresponde  galement aux attentes d'autres enseignants   qui elle pourrait servir d'exemple. Un autre int r t  tait de profiter de l'exp rience de terrain d'enseignants d j  en place pour apporter un regard p dagogique sur les contenus et la progression que l'on proposerait. Enfin cela nous permettait de pouvoir tester la s quence en classe directement. Ces trois  l ments – coh rence par rapport aux attentes, expertise p dagogique d'acteurs de terrain conjugu e   notre expertise disciplinaire et validation par des tests en classe – permettent de gagner en qualit  et en cr dibilit  dans l'offre de formations.

Depuis plusieurs ann es, des  tudiants en informatique   l'EPL¹⁵ allaient r guli rement au Lyc e Martin V (LMV), une  cole secondaire voisine, pour donner des initiations   l'informatique de deux heures dans des classes de deuxi me secondaire. Sur base de cet int r t  tabli pour l'informatique et en faisant valoir nos projets de cr ation d'une s quence de cours, nous avons pu obtenir l'adh sion de trois enseignants dans cette aventure. Nous avons donc travaill  avec deux enseignants de sciences (un homme et une femme) et un enseignant d' ducation par la technologie (EPT). A eux trois, ils ont des heures dans le cadre des activit s compl mentaires en Sciences et  ducation technologique. Ils avaient d j  l'habitude de travailler ensemble pour faire des projets-d fis bas s sur des comp tences de base du cours de sciences. Ils ont pu tester la s quence de cours dans trois classes diff rentes de la deuxi me ann e du secondaire avec $21 + 21 + 24 = 66$  l ves (entre 13 et 14 ans).

Les contraintes li es   ce contexte  taient que la s quence soit calibr e pour 10 p riodes de 50 minutes, que les activit s soient donn es en autonomie par les enseignants et que le cours s'adresse   de parfaits novices en informatique. Ce contexte  tait proche de ce que l'on peut esp rer dans d'autres  tablissements scolaires, il nous convenait donc parfaitement pour cette exp rience.

3 M thodologie

Afin de bien comprendre les conditions dans lesquelles cette s quence de cours a pu se mettre en place, nous retracons ici une chronologie des diff rents  l ments qui y ont men .

Le LMV  tait d j  sensibilis    l'informatique, mais les enseignants  taient rest s observateurs d'activit s propos es et anim es par d'autres. Un premier contact en novembre 2016 avec la direction et plusieurs enseignants a  t  pris pour pr senter notre projet. Un extrait de PV de la r union mentionne « une petite initiation   l'informatique au-del  de ce qui  tait propos  pr c demment, soit   travers de modules de quelques heures autour de trois axes : repr sentation des donn es, algorithmes, programmation par bloc ». La direction a soutenu le projet et propos    des membres de l' quipe p dagogique de participer   l'initiative sur base volontaire.

En juin 2017, il est convenu qu'une apr s-midi de formation avec quatre enseignants (sciences, EPT et art) aurait lieu en septembre. Les enseignants exposent alors le genre de projet qu'ils esp rent pouvoir r aliser avec leurs  l ves. Un consensus est rapidement  tabli vers une r utilisation interdisciplinaire des apprentissages en informatique. Les objectifs poursuivis dans la s ance de septembre  taient multiples. A ce stade, aucun des quatre enseignants n'a d'exp rience avec la programmation ou l'informatique, il fallait donc leur faire d couvrir le contenu en lui-m me. Il a  t  d cid  de leur faire vivre l'activit  « comme les  l ves », parfois en acc l r , avec une prise de recul didactique sur les activit s v cues. Le tout devait permettre aux enseignants de faire une s lection et d'agencer leur propre s quence de cours.

Cette demi-journ e de formation f t l'occasion de pr senter une s rie d'activit s d branch es permettant d'introduire quelques notions d'informatique, une  bauche de projet autour des leviers en sciences¹⁶, le logiciel Scratch et des concepts de base de la programmation. Les enseignants ont de plus  t  amen s   travailler la programmation sur machine   partir des progressions propos es sur le site code.org¹⁷. En outre, une liste de conseils didactiques tir s principalement de la litt rature en sciences de l' ducation est aussi propos e aux enseignants. A ce stade, pour des raisons personnelles, la professeure d'art doit quitter le projet.

¹⁵ Ecole Polytechnique de Louvain, Universit  catholique de Louvain

¹⁶ <https://scratch.mit.edu/projects/174396079/>

¹⁷ <https://studio.code.org/s/course3>

Cette formation a d bouch  sur la d finition des objectifs principaux de la s quence de cours   cr er. Ceux-ci sont principalement li s   la « programmation » vu l'attrait des  l ves pour cette activit  et la versatilit  que des comp tences de base en programmation am nent   l' l ve en lui permettant notamment d' tre cr atif et de r investir celles-ci rapidement vers des applications concr tes dans un contexte interdisciplinaire ( l ment critique dans le contexte restreint du syst me  ducatif belge o  l'informatique n'appara t pas explicitement dans le curriculum). En outre, dans un souci de coh rence, d'autres th mes ont  t  abord s pour introduire les raisons d' tre de la programmation. La s quence am ne donc l'apprenant   acqu rir une id e g n rale de ce qu'est un ordinateur et   d couvrir pourquoi puis comment le programmer en suivant les objectifs du r f rentiel de SI². Ces deux  l ments permettent de se construire un mod le mental du fonctionnement de cette machine articulant le c t  mat riel et le c t  logiciel. Ce mod le conduit alors   d mystifier la « magie » de l'informatique. Le dernier objectif visait   permettre   l' l ve de d velopper lui-m me un projet innovant pour se sentir acteur et non juste consommateur des outils num riques.

A partir de ce moment et jusque fin octobre quand la s quence d bute en classe et m me jusqu'  la fin de la s quence en d cembre, les enseignants ont travaill  en semi-autonomie   la production des notes de cours et   l'agencement des activit s sur base de ressources propos es. Les  changes se feront alors par emails. A partir du d but de la s quence en classe, l'auteur a observ  ponctuellement les enseignants en classe.

La s quence finale, telle que pr sent e dans les classes, est divis e en cinq chapitres. Chacun aborde un th me principal au travers d'activit s et exercices. L'ensemble de la s quence est planifi  pour dix « heures » de cours par blocs de deux p riodes cons cutives (2 fois 50 minutes), elle prendra finalement un peu plus de temps. En fin de s quence, une  valuation sommative a  t  propos e par l'auteur. En effet, le syst me  ducatif n cessite l' valuation certificative des acquis d'apprentissage traduite par une note et les enseignants souhaitaient avoir un regard ext rieur sur le niveau atteint au terme de la s quence. Au sein de notre exp rience, cette  valuation est une mesure de la compr hension g n rale de la classe par rapport aux activit s propos es. Cette  valuation s'est enti rement d roul e sur papier et n'a donc pas eu de modalit  sur machine.

Chapitre 1 : Un ordinateur, cette machine stupide

Objectif : Comprendre le r le des diff rents composants d'un ordinateur dans l'ex cution d'un programme

D roul  : Ax  dans un premier temps autour des composants et de l'architecture de l'ordinateur et dans un deuxi me temps autour de ses capacit s de traitement de l'information, ce chapitre fait  merger la repr sentation d'une machine qui ne fait que traiter des donn es de fa on formelle en passant par leur repr sentation num rique.

Activit  1 : Les composants de l'ordinateur : Faire  merger les repr sentations des  l ves, puis les enrichir en les confrontant   diff rentes images et   la d couverte de l'int rieur d'une tour d'ordinateur pour aboutir   un sch ma d'architecture simplifi .

Activit  2 : Les limites de l'ordinateur : Activit  sur le traitement formel inspir e de visaTICE¹⁸, puis  valuer le niveau de difficult  de t ches   effectuer sur un texte dans une langue  trang re comme analogie   un ordinateur face   de l'information num ris e (e.g. « Dire combien il compte de lettres » : t che facile ou « Nombre de noms communs » : t che difficile).

Structuration visant   associer des termes (composant ou concept) avec leur d finition et aboutir   une d finition de classe de l'ordinateur.

Chapitre 2 : Non ce n'est pas de la magie (inspir e du site de Paul Curzon CS4fn¹⁹)

Objectif : Lire et comprendre un algorithme simple permettant de r soudre un probl me donn .

D roul  : Un  l ve, *la m moire*, lit une suite d'instructions   un deuxi me, *le processeur*, qui les r alise. Un troisi me, *l'utilisateur*, doit intervenir en fonction des instructions du processeur. Cette activit  fait r aliser aux  l ves un tour de magie et met en  vidence qu'aucun des trois acteurs ne doit vraiment comprendre le tour pour peu que les instructions soient suffisamment claires. L'intelligence n'est pas dans l'ordinateur mais dans la programmation de ces instructions.

Chapitre 3 : L'algorithme de la tartine (inspir e de CS50   Harvard²⁰)

Objectif : Concevoir un algorithme simple permettant de r soudre un probl me donn 

D roul  : Les  l ves doivent donner des instructions   leur professeur devenu un robot. La t che   lui faire r aliser consiste   *faire une tartine au beurre et   la confiture*. Pour la premi re fois, les  l ves doivent  crire eux-m mes

¹⁸ visatice.ulg.ac.be

¹⁹ <http://www.cs4fn.org/mathemagic/>

²⁰ <https://www.youtube.com/watch?v=KUB-aJXquUA&feature=youtu.be&t=18m>

des instructions pr cises pour faire faire quelque chose   un syst me qui prend les instructions au pied de la lettre. La structuration fera  merger les notions d'instruction univoque et de granularit .

Chapitre 4 : Cherchons le tr sor

Objectif : D finir des instructions non ambigu s

D roul  : Une succession de grilles sont propos es pour lesquelles l' l ve doit diriger un personnage d'une position initiale jusqu'  un objectif. Dans un premier temps, la classe d finit un jeu d'instructions, un langage, qui permettra de diriger le robot. Sur base de ces instructions, cinq situations progressivement plus complexes am nent des concepts tels que les boucles, conditions, fonctions.

Chapitre 5 : Programmation par blocs

Objectif : Comprendre qu'un langage de programmation est une fa on de d crire un algorithme qui doit  tre compr hensible par une machine, et exercer sa capacit    imaginer des algorithmes simples et   les impl menter.

D roul  : Ce chapitre prend   lui seul la moiti  du temps estim  de la s quence et se fait sur ordinateur. En suivant une progression propos e sur le site de code.org, les  l ves ont une s rie de petits d fis   r aliser dans une interface de programmation par blocs. Ils doivent imbriquer plusieurs blocs visuels qui repr sentent les diverses instructions qui constituent un programme pour diriger un personnage, soit dans une grille avec un objectif, soit comme la tortue logo dans le but de dessiner des formes g om triques. On notera que pour les exercices sur machine, les  l ves travaillent par paire en changeant de programmeur   chaque puzzle.

4 R sultats et discussion

Parmi les r sultats obtenus, il est d'abord int ressant de pr senter ceux de l' valuation sommative qui a  t  propos e aux  l ves. On peut observer que m me si la majorit  d'entre eux a obtenu plus de la moiti , ce sont les questions sur le concept de fonction et sur la correction d'un programme qui ont caus  le plus de soucis. Sans doute pour le premier car c'est le concept le plus abstrait abord  et pour le second parce qu'il avait  t  moins travaill  pendant la s quence.

Tableau 1. R sultats par groupe   l' valuation sommative de fin de s quence

	Groupe 1 (n=24)	Groupe 2 (n=21)	Groupe 3 (n=21)
Taux de r�ussite au th�me « fonction »	25	81	-
Taux de r�ussite au th�me « correction de programme »	70	66	-
Taux d'�chec global (<50%)	33	10	5
Taux entre 50% et 70 %	33	33	24
Taux entre 70% et 100%	33	57	71
Taux de r�ussite (>=50%)	66	90	95

L'avis des trois enseignants  tait tr s positif vis- -vis de l'exp rience. Leur motivation du d but   la fin a  t  au rendez-vous. Quant   savoir s'ils sont pr ts   r it rer le projet l'ann e prochaine, ils se montrent m me impatients de pouvoir redonner la s quence en connaissance de cause afin de pouvoir am liorer et corriger les d fauts observ s et surmonter les erreurs de jeunesse. Confort s dans leur volont  de vouloir utiliser l'informatique dans le cadre de leurs cours, ils ont confirm  leur intention d'organiser une deuxi me s quence au second semestre bas e sur la r alisation d'un projet en Scratch autour du cours de sciences et des notions de masse volumique. Cela permettra aux  l ves de r investir leurs nouvelles comp tences dans un cadre plus cr atif et interdisciplinaire.

Les enseignants ont soulign  quelques aspects lors d'une rencontre a posteriori :

- un grand plaisir   travailler en  quipe sur un nouveau projet qui leur demandait d'apprendre de nouveaux contenus li s   l'informatique et de se coordonner ;
 - un inconfort relatif dans l'id e d'apprendre quasi en m me temps que leurs  l ves, m me s'ils ont pu le g rer malgr  un manque de vision globale de la s quence et des attendus finaux ;
 - un d but de sentiment d' tre des enseignants d'informatique m me s'ils ne se sentaient pas encore des experts.
- Ces diff rents aspects sont tr s positifs et d j  observ s dans d'autres exp riences similaires comme le souligne Mark Guzdial²¹. En notant toutefois qu'on a travaill  ici avec des enseignants plut t enthousiastes au sens de Roche, De La Higuera, & Michaut (2018).

²¹ <https://computinged.wordpress.com/2016/05/23/what-is-the-community-of-practice-for-cs4all-teachers-suggestion-its-not-teachers/>

Lors des observations en classe, plusieurs types d'erreurs ont  t  observ es :

- Un manque de rigueur de la part d'un enseignant : e.g. pr senter une instruction conditionnelle comme si elle  tait implicitement dans une boucle infinie avec la condition comme condition d'arr t.
- Un d sarroi vis- -vis de certains comportements d' l ves : un  l ve ne comprend pas pourquoi on s'obstine   vouloir prendre le chemin le plus court sur la grille.
- Un probl me devant la gestion des diff rences de niveaux en classe (notamment parce que pr s de 30% des  l ves dans une classe avaient d j  manipul  la programmation par blocs en primaire). A ce sujet, plusieurs pistes sont propos es en r union de d briefing, comme envoyer les  l ves les plus rapides aider les autres ou encore encourager les  l ves devant une interrogation   s'aider mutuellement avant d'interroger l'enseignant.

Ceci montre d'une part que les enseignants ont besoin d'une formation plus approfondie, avec notamment des aspects didactiques et de la pratique si on attend d'eux qu'ils puissent g rer les cours en autonomie. D'autre part, certaines observations sont g n riques et donc pas propres   l'informatique (notamment la diff renciation).

Comme attendu, l'exp rience en classe a permis de corriger voire m me de supprimer un certain nombre de contenus inadapt s ou d'activit s mal d finies. A titre d'exemple, les instructions pour l'activit  « Non, ce n'est pas de la magie » ont  t  jug es parfois trop impr cises ou pas assez claires par les  l ves comme par les enseignants. L'activit  sur les composants de l'ordinateur a pris du temps et a trouv  peu de sens dans le contexte de la s quence de cours. Dans l'activit  « les limites de l'ordinateur », l'option qui permettait un choix interm diaire entre « difficile » ou « facile » est supprim e. Sur base de ces diff rents retours et observations, des modifications ont  t  apport es aux notes de cours. La version finale a  t  retravaill e et est d sormais disponible pour d'autres enseignants qui voudraient tester la s quence avec leurs  l ves²².

Un autre r sultat de cette exp rience est que le produit de cette collaboration a d j   t  propos  plusieurs fois dans le cadre de formations continues   d'autres enseignants. Parmi ceux-ci, plusieurs y ont trouv  des informations et des ressources d'int r t. La s quence est en g n ral consid r e comme abordable pour tout novice. Des professeurs de Fran ais envisageaient de r utiliser les activit s d branch es qui portaient sur la pr cision des instructions dans le cadre de leurs cours pour travailler la notion d' criture et lecture de consignes. La plupart des enseignants ont cependant manifester une peur devant la mati re   pr senter pour laquelle ils ne se sentaient pas experts et un manque de temps pour int grer la s quence dans leur programme de cours.

5 Conclusion

L'apport principal de ce retour d'exp rience est d'avoir montr  qu'il est possible de profiter de l'exp rience de terrain de professeurs pour cr er et valider une s quence de cours en informatique. Des enseignants totalement d butants ont pu g rer les activit s apr s un nombre restreint d'heures de formation et avec un suivi longitudinal durant l'ann e. Les avis  taient globalement positifs, ils sont all s au bout de ce qu'ils avaient annonc  et malgr  quelques erreurs de contenu ou de gestion de certaines activit s, les objectifs fix s sont atteints.

Les  l ves ont  t  r ceptifs m me si les concepts les plus abstraits ne sont pas ma tris s par tous. Mais ce n'est pas  tonnant avec seulement 10 p riodes de 50 minutes consacr es   cette nouvelle discipline.

La d marche sera renouvel e l'ann e prochaine au LMV avec l'appui de la direction et SP² compte bien  tendre l'exp rience dans d'autres  coles avec des mesures moins normatives, tant pour ce qui est de la formation des enseignants que pour la mesure des apprentissages des  l ves.

On notera encore qu'une deuxi me s quence est pr vue au second semestre dans ces trois classes pour remobiliser les concepts de programmation dans le cadre d'un projet interdisciplinaire. Les r sultats de cette nouvelle activit  seront  galement dignes d'int r t.

R f rences

Bersini, H. et al. (2018) *Rapport sur l'impl mentation d'un tronc commun dans l'enseignement obligatoire*. Rapport de l'Acad mie Royale

²² https://sicarre.be/pmwiki/uploads/Documents/introduction_programmation.pdf

Brown, N. C., Sentance, S., Crick, T., & Humphreys, S. (2014). Restart: The resurgence of computer science in UK schools. *ACM Transactions on Computing Education (TOCE)*, 14(2), 9.

Caspersen, M. E., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2018). Informatics for All The strategy.

FESeC (2007). *Activit s compl mentaires pour le 1er degr  : EPT : INFORMATIQUE*. D/2007/7362/3/35

Gal-Ezer, J., & Stephenson, C. (2014). A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. *Trans. Comput. Educ.*, 14(2), 8:1–8:18. <https://doi.org/10.1145/2602483>

Gander, W., Petit, A., Berry, G., Demo, B., Vahrenhold, J., McGettrick, A., ... Ghezzi, C. (2013). Informatics education: Europe cannot afford to miss the boat. *Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education*.

Henry, J., & Smal, A. (2018). «Et si demain je devais enseigner l'informatique?» Le cas des enseignants de Belgique francophone. In *Didapro 7–DidASTIC. De 0   1 ou l'heure de l'informatique   l' cole*.

Roche, M., De La Higuera, C., & Michaut, C. (2018). Enseigner la programmation informatique: comment r agissent les professeurs des  coles? *Notes du CREN*, (27).

SI (2017). *Pourquoi pensons-nous que l'informatique a sa place dans l'enseignement obligatoire ?*