

École polytechnique de Louvain

Didactique de l'informatique : Élaboration de séquences de cours pour des élèves de primaire

Auteur: **Charline OUTTERS**

Promoteur: **Kim MENS**

Lecteurs: **Xavier GILLARD, Olivier GOLETTI, Chantal PONCIN**

Année académique 2018–2019

Master [60] en sciences informatiques

Résumé

Le numérique est de plus en plus présent dans le monde actuel, il est donc important que la formation de la jeunesse en tienne compte. Pourtant, en Wallonie, les jeunes terminant l'enseignement obligatoire n'ont généralement aucune formation numérique, sauf peut-être, l'utilisation d'un ordinateur et de logiciels de bureautique. Ceci est en voie de changer avec la réforme du tronc commun qui entrera en vigueur dans les années à venir.

Nous avons élaboré deux séquences de cours pour initier les jeunes de fin de primaire à l'algorithmique et à la programmation. L'une de ces séquences porte sur la robotique et a pu être testée dans une classe de 5^{ème} primaire. Cela a permis de mettre en lumière certaines difficultés inattendues rencontrées par les enfants, de réfléchir à l'origine de celles-ci et de retravailler les activités en fonction. La deuxième séquence, qui n'a pu être testée avec les enfants, dresse un parallèle entre les algorithmes et les recettes de cuisine.

Remerciements

Merci au Professeur Kim Mens d'avoir accepté d'être mon promoteur.

Merci également à Chantal Poncin et Olivier Galletti pour leur supervision tout au long de ce mémoire, pour leurs commentaires et le temps qu'ils m'ont accordé.

Merci à Bérengère Daxhelet et à ses élèves de 5ème primaire de m'avoir accueillie dans leur classe.

Merci aussi à Xavier Gillard pour l'intérêt porté à ce projet.

Enfin, merci à ma maman pour sa relecture attentive. Merci à mes frères, mon papa, Jérémie, et, mes amis et mes colocs, Caro, Vu, Julien, Arnaud, Antoine, Robin, Romain, Corentin, Kevin, François, Tanguy, Pat, Alexandre, Nico, Tom, Alexis, Charlotte, Gillou, Élisabeth, Élise, Benjamin et Sylvain, d'avoir embelli ces dernières années et d'avoir été une source d'inspiration quand j'en manquais.

Table des matières

Introduction	1
1 Etat de l'art	2
1.1 Situation en Belgique et ailleurs	2
1.2 Concepts nécessaires à une initiation à l'informatique	4
2 Robotique et programmation événementielle	6
2.1 Objectifs et ressources	6
2.1.1 Objectifs de la séquence	6
2.1.2 Qu'est-ce qu'un robot ?	7
2.1.3 Ressources	8
2.2 Présentation de la séquence	11
Tableau récapitulatif de la séquence	11
2.2.1 Séance 1 - Introduction au robot	15
2.2.2 Séance 2 - Observer les comportements de Thymio	15
2.2.3 Séance 3 - Formalisation des comportements de Thymio	16
2.2.4 Séance 4 - Utilisation des différents comportements	17
2.2.5 Séance 5 - Introduction à la programmation VPL	18
2.2.6 Séance 6 - Programmation simple de Thymio	20
2.2.7 Séance 7 - Programmes un peu plus compliqués	21
2.2.8 Séance 8 - Résolution de notre problématique de départ	22
2.2.9 Séance 9 - Quiz final	23
2.2.10 Séance 10 - Programmation VPL avancée - Introduction	24
2.2.11 Séance 11 - Programmation VPL avancée - Suite	24
2.2.12 Séance 12 - Thymio et le code de la route	25
2.3 Retours et améliorations	25
2.3.1 Déroulement des séances	25
2.3.2 Améliorations	31
2.3.3 Analyse	31
2.4 Points d'attention et recommandations pour les enseignants	32
3 Algorithmes et recettes de cuisine	35
3.1 Objectifs et ressources	35
3.1.1 Objectifs de la séquence	35
3.1.2 Ressources	36
3.2 Présentation de la séquence	39
Tableau récapitulatif de la séquence	39

3.2.1	Séance 1 - La précision des instructions	43
3.2.2	Séance 2 - Les instructions induites	44
3.2.3	Séance 3 - Les conditions	45
3.2.4	Séance 4 - Les boucles	46
3.2.5	Séance 5 - Les variables	47
3.3	Discussion	48
Conclusion		49
A Formalisation du comportement jaune		52
B Exercice de débbugging		53
C Recette du dessert aux framboises		54
D Tâche d'initiation à INGIInious		55
E Arbre de décision		56
F Tâche blockly : le tri des élèves		57
G Recette du quatre-quart		58

Table des figures

2.1	Boucle sensori-motrice (Source : le MOOC <i>Le robot Thymio comme outil de découverte des sciences du numérique</i> sur www.fun-mooc.fr)	8
2.2	Composition d'un Wireless Thymio (Source : https://www.thymio.org/)	9
2.3	Les différents pictogrammes du langage VPL	11
2.4	Contenu du set de pictogrammes à adapter reçu par les enfants	17
2.5	Interface d'Aseba VPL	19
2.6	Feedback visuel offert par la programmation VPL (avec la flèche jaune, l'instruction en train d'être exécutée)	19
2.7	Exemple d'exercice d'enchaînement d'instructions	20
2.8	Exemple d'exercice de vrai ou faux	21
2.9	Exemple d'exercice de prédiction	22
3.1	Présentation d'un exercice Blockly sur la plateforme INGINIOUS	37
3.2	Exemples de blocs créés sur mesure	38
3.3	Les différents espaces d'une tâche Blockly	39
3.4	Exemple d'images reçues par les enfants (le gâteau correct est en rose et le faux en vert)	45
A.1	A gauche, la feuille reçue par les élèves et à droite la formalisation attendue	52
D.1	Au dessus, le résultat attendu avec le message de succès et en dessous, le message d'erreur.	55
F.1	Résultat attendu pour la tâche de tri des élèves	57

Liste des tableaux

2.1	Récapitulatif de toutes les activités pensées pour la séquence sur la robotique	14
2.2	Activités sélectionnées pour les séances en classe	26
2.3	Activités finalement réalisées en classe	30
2.4	Séance améliorée selon les observations réalisées	34
3.1	Récapitulatif des activités prévues dans la séquence sur l'algorithmique . .	42

Introduction

Le numérique prend tous les jours un peu plus de place dans le monde dans lequel nous vivons et nous y sommes confrontés dès le plus jeune âge. C'est pourquoi, il est crucial que les jeunes reçoivent une formation dans ce domaine, afin de pouvoir appréhender ce monde toujours plus connecté de la manière la plus pertinente et sécurisée possible.

La Wallonie était très mauvaise élève en ce domaine en comparaison avec d'autres pays, mais c'est en passe de changer grâce au Pacte pour un Enseignement d'Excellence qui inclut de nouvelles compétences en sciences informatiques. Le premier chapitre de ce mémoire explique les changements qui vont être mis en place, ainsi que les objectifs poursuivis par l'intégration de ces compétences. Il fait également un tour d'horizon de ce qui se fait en la matière dans d'autres pays.

Ce mémoire a pour but de proposer deux séquences de cours pouvant être données en fin de primaire afin d'initier les élèves à l'informatique et de tester l'une d'elles avec des enfants. Ces séquences sont centrées sur la programmation et l'algorithmique, deux domaines clés des sciences informatiques. Le chapitre 2 présente la première séquence portant sur la robotique et ayant pour but d'initier les enfants à la programmation événementielle. Il dévoile les objectifs poursuivis, les activités prévues pour chacune des séances et la manière dont elles se sont déroulées dans la classe pour en réaliser une critique constructive et apporter des idées d'améliorations. Le dernier chapitre expose, de la même manière, la deuxième séquence qui fait le parallèle entre l'algorithmique et les recettes de cuisine. Comme elle n'a pas été réalisée dans une classe, la partie critique est remplacée par une discussion à propos des avantages d'une telle séquence.

Chapitre 1

Etat de l'art

La première section de ce chapitre présente la situation par rapport à la didactique de l'informatique en Belgique francophone. Elle développe, dans un premier temps, l'état actuel des choses en Fédération Wallonie-Bruxelles et ensuite, ce qui se fait de l'autre côté de la frontière linguistique, en Communauté flamande. Elle pointe également quelques initiatives chez nos voisins européens et sur les autres continents.

La plupart de ces initiatives sont fondées sur de grandes thématiques assez semblables à travers le monde. Ces thématiques sont expliquées dans la deuxième section de ce chapitre et une attention particulière est portée à celles en lien avec les séquences de cours développées dans le cadre de ce mémoire.

1.1 Situation en Belgique et ailleurs

A l'heure actuelle, en Belgique francophone, un jeune de 18 ans ayant terminé son enseignement obligatoire n'aura quasiment aucune formation en informatique, comme l'ont révélé J. Henry et N. Joris dans deux articles [1][2]. Elles poursuivent en indiquant qu'il existe cependant, dans certaines écoles des réseaux catholique et de la Fédération Wallonie-Bruxelles, un cours d'initiation à l'informatique. Ce cours prend la forme d'une activité complémentaire, l'enseignant qui a la charge du cours décide de ce qu'il souhaite y mettre ou non, selon la ligne directrice décidée par l'école. En Fédération Wallonie-Bruxelles, par facilité, le cours se base souvent sur les modules prévus dans le Passeport TIC¹, couvrant essentiellement le sujet de l'utilisation de l'ordinateur et des logiciels, ainsi qu'un côté plus matériel portant sur l'ordinateur et ses périphériques. Dans l'enseignement catholique, ce cours est souvent lié à celui d'Éducation par la technologie² et s'apparente malgré tout à l'apprentissage de la bureautique ("Avoir une maîtrise du clavier", "Savoir utiliser l'Internet", "Savoir utiliser un traitement de texte"). Dans l'enseignement non général, il existe des options en "sciences informatiques" ou "informatique" en technique de transition, et une option "technicien en informatique" en technique de qualification qui consiste à l'installation, à la maintenance et au dépannage d'équipements informatiques.

1. <http://www.enseignement.be/index.php?page=27182&navi=3683>

2. <http://admin.segec.be/documents/4861.pdf>

Cependant, cela devrait changer avec la réforme du tronc commun (primaires et secondaires inférieures) souhaitée par la Fédération Wallonie-Bruxelles avec son Pacte pour un Enseignement d'Excellence³. Ce pacte a pour but de diminuer le nombre de redoublement de 50% d'ici 2030, en favorisant un accompagnement plus personnalisé des élèves, afin de sortir la Belgique de sa dernière place européenne en la matière. Cette réforme entrera en vigueur à la rentrée 2020 pour les classes de maternelle et du début de primaire. Elle propose sept grands domaines d'apprentissage, dont le domaine "Mathématiques, Sciences et Techniques". Ce domaine est composé de quatre parties : "Mathématiques", "Géographie physique", "Sciences" et "Formation manuelle, technique, technologique et numérique"⁴. La formation numérique s'intéresse aux cinq aires de compétences numériques définies par la Commission Européenne [3] : culture de l'information et des données, communication et collaboration, création de contenu numérique, sécurité et enfin, résolution de problème.

Différentes universités et Hautes Écoles organisant des programmes en informatique ont voulu influencer le contenu de cette partie numérique. Ils ont créé un groupe de travail, nommé Sciences Informatiques au Secondaire Inférieur (SI²)⁵, pour réfléchir au contenu devant être abordé dans le premier degré du secondaire et établir un référentiel des compétences à acquérir. Celui-ci propose un enseignement des sciences informatiques autour de cinq thématiques inspirées des expériences effectuées à l'étranger. Ces thématiques sont la représentation des données, l'algorithmique, la programmation, le matériel et enfin, le réseau et la sécurité. Elles sont expliquées dans la section suivante.

Au Nord du pays, selon le CECE (Committee on European Computing Education) [4], les élèves rencontrent le monde numérique dès la maternelle. En primaire, les enfants exercent leurs compétences en technologie de l'information et de la communication de manière transversale⁶. Ils apprennent généralement à utiliser un ordinateur ou une tablette. Avec la réforme de l'enseignement orchestrée par la Communauté flamande, les élèves peuvent choisir une option STEM⁷ dès la troisième année de secondaire [5]. Il existe néanmoins des écoles proposant une activité complémentaire STEM dès la première année. Ces cours peuvent contenir de l'informatique mais ce n'est pas une obligation.

F. Heintz, L. Mannila et T.Färnqvist ont observé la façon d'introduire l'informatique dans divers pays [6], nous nous attardons sur les plus intéressants. En Europe, l'Angleterre se démarque en imposant un cours d'informatique en primaire. Il ne se focalise pas uniquement sur la composante de programmation, mais représente une discipline à part entière traitant des sciences informatiques, des technologies de l'information et de la culture numérique. Les objectifs de ce cours sont, entre autres, de savoir utiliser les principes d'abstraction, de logique et d'algorithmique, d'être capable de représenter les données, de pouvoir analyser des problèmes, de programmer des solutions et d'être responsable dans son utilisation des technologies d'information et de communication. En Pologne, les sciences informatiques font partie du cursus de secondaire depuis bien longtemps. Depuis

3. <http://www.pactedexcellence.be/>

4. https://www.fondation-enseignement.be/sites/default/files/FPE/pdf/PPT%20Fondation%20pour%20l%27enseignement%20VdefLM%20MMSCHYNS_0.pdf

5. <https://sicarre.be/>

6. <https://www.onderwijskiezer.be/>

7. Acronyme utilisé pour désigner quatre disciplines : science, technologie, ingénierie et mathématiques.

1999, il existe une formation en sciences informatiques répartie sur les années de primaire et de secondaire. En primaire l'intérêt principal est porté aux technologies de l'information et de la communication. Dans le cycle inférieur du secondaire, les élèves sont amenés à faire de l'algorithmique et de la résolution de problèmes. Et enfin, dans le cycle supérieur, la programmation n'est pas obligatoire, mais elle est généralement introduite par les écoles.

Dans le monde, les Australiens de 6 à 18 ans suivent un cursus sur les technologies dont celles du numérique, où ils apprennent, entre autres, les données, les systèmes numériques et la programmation. Et enfin, aux États-Unis, les élèves de secondaire ont le choix de suivre un cours sur les principes des sciences informatiques et de passer un examen national sur le sujet. Ce cours leur apprend à reconnaître et représenter des problèmes informatiques, à développer et utiliser des abstractions, à créer une culture informatique inclusive, ...

1.2 Concepts nécessaires à une initiation à l'informatique

Plusieurs raisons sont mises en avant par M. Guzdial pour expliquer l'avantage d'apprendre les sciences informatiques au sens large dans les écoles [7]. Premièrement les débouchés, car de plus en plus de métiers nécessitent une connaissance en sciences informatiques ou sont révolutionnées par ces dernières (par exemple, les statistiques depuis l'avènement du machine learning). Ensuite, cela permet de mieux comprendre le monde numérique dans lequel nous évoluons et de mieux utiliser les outils à notre disposition. Cet apprentissage permettrait aussi de vaincre certains stéréotypes en suscitant des vocations auprès des filles, par exemple. Une dernière raison évoquée par M. Guzdial est celle visant à initier la population à la pensée informatique. Cette dernière raison est le moteur de notre travail dans ce mémoire.

La pensée informatique n'a pas une définition unique, mais la description que J. Wing en a fait [8] semble être une base commune à toutes ces définitions. Selon J. Wing, la pensée informatique est le cheminement cognitif utilisant les principes fondamentaux des sciences informatiques (abstraction, représentation des données, logique, décomposition de problèmes en problèmes résolubles, algorithmique,...) afin de résoudre des problèmes, concevoir des systèmes et même, comprendre le comportement humain. Elle perçoit deux gains à développer ce type de pensée : l'application d'idées développées dans ce domaine pour faciliter le travail informatique dans d'autres domaines, mais aussi, appliquer ces idées dans notre vie quotidienne afin de résoudre des problèmes tels que choisir la bonne file à la caisse du supermarché, ce qui s'apparente à la modélisation des performances d'un système multi-serveurs. Dès lors, nos séquences de cours d'introduction aux sciences informatiques sont également une initiation à la pensée informatique.

Dans son référentiel, qui est notre référence pour ce mémoire, le groupe de travail SI² [9] a identifié cinq axes permettant de découvrir les compétences fondamentales à développer lors d'une introduction aux sciences informatiques.

- **La représentation des données** : visant à comprendre comment un ordinateur ne fonctionnant que sur base de 0 et de 1, fait pour représenter les images, les

caractères des textes et les nombres.

- **Les réseaux et la sécurité** : visant à donner les outils nécessaires pour comprendre le fonctionnement des réseaux et les problèmes inhérents.
- **Le matériel** : visant à identifier les composants de base d'un ordinateur pour comprendre comment il peut fonctionner dans les grandes lignes.
- **L'algorithmique** : visant à ce que les élèves réalisent qu'une solution doit être exprimée par étape, à l'aide de diverses instructions, et qu'ils soient capables de lire et composer des algorithmes ainsi que de discuter de leur efficacité.
- **La programmation** : visant à traduire des algorithmes dans un langage compris par la machine.

Parmi les cinq axes, deux nous intéressent particulièrement car ils sont explorés dans les séquences de cours créées dans le cadre de ce mémoire.

En algorithmique, les enfants seront amenés à lire, comprendre, concevoir et utiliser des algorithmes simples permettant de résoudre des problèmes donnés. Pour arriver à ce résultat, ils devront être capables d'adapter ou de simuler l'exécution de certains algorithmes. Ils devront également pouvoir identifier et utiliser des instructions existantes, mais aussi en créer de nouvelles en veillant à ce qu'elles ne soient pas ambiguës. Et enfin, pouvoir combiner ces dernières à des structures de contrôle (boucles, conditions) et des variables, de manière à concevoir des algorithmes valides.

L'axe de la programmation est souvent travaillé en parallèle avec celui de l'algorithmique, car un programme peut être considéré comme la traduction d'un algorithme dans un langage compréhensible par une machine. Les enfants apprendront à rédiger des programmes sur base ou non d'un algorithme existant, mais ils seront également amenés à corriger ce programme après avoir établi les cas de base à tester et avoir vérifié si le programme produit les résultats attendus. Il existe différents types de programmation, les séquences élaborées en travaillent deux :

- **La programmation séquentielle** dans laquelle les instructions se déroulent les unes après les autres dans le même ordre.
- **La programmation événementielle** dans laquelle les instructions exécutées, appelées actions, dépendent d'événements extérieurs. Par conséquent, les instructions peuvent se dérouler dans un ordre différent selon l'environnement dans lequel le programme est exécuté.

Ces différents types de programmation fonctionnent tous sur base d'instructions et utilisent des structures de contrôle, à savoir les boucles et les conditions, de manière plus ou moins explicite. Chaque type de programmation a des avantages le rendant plus efficace dans certains domaines que d'autres. La programmation séquentielle est très utilisée pour faire des calculs. Par exemple, le programme reçoit des données de base et calcule un résultat selon différentes contraintes. Peu importe le nombre de fois où le programme est exécuté, il donnera toujours le même résultat avec les mêmes données de départ. La programmation événementielle, quant à elle, est plus performante en robotique puisqu'elle privilégie l'interaction avec l'environnement. Il est donc plus facile de programmer une réaction lors de la perception d'un obstacle dans ce type de programmation.

Chapitre 2

Robotique et programmation événementielle

Cette première séquence de cours aborde le sujet de la robotique. Les objectifs poursuivis et les ressources utilisées sont développées dans la première section de ce chapitre. Ensuite, la séquence de cours telle qu'elle avait été pensée, est exposée en détail et est suivie par une section expliquant les observations effectuées lors de l'expérimentation en classe. Enfin, une conclusion reprend les différents points d'attention ainsi que des recommandations destinés aux enseignants qui voudraient proposer cette même séquence de cours.

2.1 Objectifs et ressources

Cette section décrit les objectifs poursuivis par cette séquence et en quoi ils s'intègrent aux grands axes définis dans la section 1.2. Ensuite, le concept de robot est explicité ainsi que toutes les notions qui y sont liées. Et pour finir, les ressources utilisées dans la séquence sont présentées et justifiées.

2.1.1 Objectifs de la séquence

Pour cette première séance, le choix de travailler avec des robots s'est imposé grâce à son côté fascinant aux yeux des enfants, comme le dit l'équipe de S. Magnenat [10][11][12]. Les enfants étant tout de suite très enthousiastes et curieux à l'égard des robots, ils sont parfois plus motivés à travailler [10]. La deuxième question importante se posant, après le choix de travailler avec le robot, est de décider si l'on veut faire une séquence éducative à l'aide du robot (c'est-à-dire utiliser le robot pour faire des mathématiques ou autres), ou si l'on souhaite faire une séquence éducative à propos du robot. Nous avons opté pour la deuxième solution, cependant, certaines compétences des cours de français et autres sont également travaillées.

Le but principal de cette séquence est donc de faire découvrir plusieurs concepts de la robotique et de l'informatique aux enfants. Le projet IniRobot¹, qui développe un kit

1. Projet initié par l'équipe Flowers d'INRIA (Institut National de Recherche en Informatique et en Automatique) en 2014 qui rassemble une série d'activités pédagogiques autour de la robotique et de la programmation en primaire

d'initiation pour ce domaine, vise ce même objectif et a dressé une liste des compétences à travailler pour y parvenir[13]. Cette séquence travaille ces mêmes compétences :

- Comprendre, connaître et pouvoir utiliser les notions de "capteurs", "actionneurs", "informatique", "programme", "instruction", "électronique", "mécanique", "algorithme" et "langage de programmation" ;
- Comprendre la boucle sensori-motrice et savoir qu'elle est composée de capteurs, d'actionneurs et d'un ordinateur ;
- Comprendre que le comportement du robot dépend de l'interaction entre un programme, le corps du robot et l'environnement ;
- Connaître les concepts de base de la programmation événementielle (action, événement) et comment utiliser les règles des " si... alors... " ;
- Comprendre comment donner des instructions à un robot et savoir qu'un algorithme est composé d'une suite d'instructions ;
- Pouvoir faire une analogie entre un être vivant et un robot.

Le développement de ces compétences robotiques implique le travail de certaines compétences scolaires présentées dans les Socles de compétences [14]. C'est le cas du travail en équipe, de la décomposition de problèmes en plus petites entités, ou encore, de la formalisation d'observations.

2.1.2 Qu'est-ce qu'un robot ?

Pour présenter l'élément principal de cette séquence, à savoir le robot, nous allons commencer par en donner une définition, puis expliquer les différentes notions qui en découlent comme présentées dans la section précédente. Un robot est donc une machine alliant **mécanique**, **électronique** et **informatique**, capable de percevoir son environnement et de réagir en fonction, grâce au travail conjoint de **capteurs**, d'**actionneurs** et d'un ordinateur.

La robotique est une alliance de trois domaines.

- La mécanique : domaine qui est en charge des mouvements du robot et de tous les mécanismes internes qui les permettent ;
- L'informatique : domaine qui est en charge du traitement des informations perçues par le robot, grâce à un programme ;
- L'électronique : domaine qui est en charge de tous les composants électriques nécessaires pour que les deux domaines précédents fonctionnent et pour que le robot puisse percevoir son environnement à l'aide de ses capteurs.

À cela s'ajoutent les notions de capteur et d'actionneur qui sont essentielles au concept de boucle sensori-motrice. Les capteurs sont des dispositifs capables de transformer une grandeur physique (température, distance, pression, ...) en signal, c'est-à-dire en information pouvant être traitée par le programme informatique du robot. Les actionneurs sont les mécanismes qui permettent d'agir sur le robot en fonction des instructions données par le programme, en vue de modifier son comportement ou son état. La boucle sensori-motrice est représentée dans l'image de la figure 2.1. Le robot perçoit la présence de la balle grâce à son capteur frontal. Cette information est ensuite transmise à l'ordinateur qui la traite

et ordonne à l'actionneur de faire quelque chose, le robot fait donc une action.

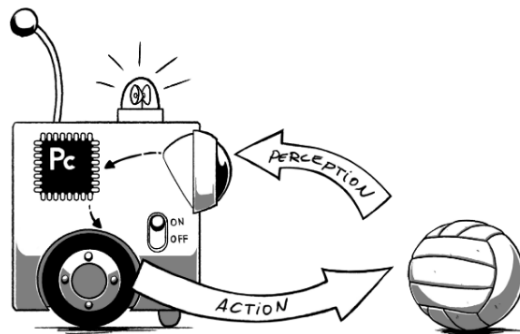


FIGURE 2.1 – Boucle sensori-motrice (Source : le MOOC *Le robot Thymio comme outil de découverte des sciences du numérique* sur www.fun-mooc.fr)

2.1.3 Ressources

Le robot

Une fois les objectifs de la séquence de cours établis, il a fallu déterminer le robot le plus propice pour la réaliser. Notre choix s'est porté sur le robot Thymio II² inclu dans le projet Ecole Numérique³ dans le cadre de Digital Wallonia⁴. Thymio II est un robot éducatif distribué par Mobsya, une association sans but lucratif. Le projet a démarré suite à une collaboration entre l'École Polytechnique Fédérale de Lausanne (EPFL) et l'École Cantonale d'Art de Lausanne (écal). Ensuite, de multiples acteurs (instituts de recherche, enseignants, designers) se sont liés au projet afin d'en faire une plateforme d'autant plus adaptée à l'apprentissage.

Thymio a pour but la découverte et l'apprentissage de la pensée informatique qui a été développée dans le chapitre précédent, section 1.2. Le souhait de ses créateurs est qu'il soit le plus accessible possible, c'est pourquoi il est distribué en open source⁵ tant au niveau matériel que logiciel. Cela a pour avantage d'avoir une communauté très active autour du robot qui crée et partage des activités, des cours, des exercices, ... Un autre avantage de Thymio lié à son accessibilité est son design. En effet, il est très neutre esthétiquement contrairement aux LEGO Mindstorms qui ont des designs bien plus attirants auprès des garçons. Comme il s'agit ici de donner des cours dans une classe mixte, il est important que filles et garçons puissent se sentir concernés.

Thymio est un petit robot mobile d'une dizaine de centimètres qui possède énormément de capteurs et d'actionneurs, pourtant il est distribué à un prix abordable (une grosse centaine d'euros), toujours dans cette optique d'être accessible au plus grand nombre. Les

2. <https://www.thymio.org/fr/>

3. <http://www.ecolenumerique.be/qa/>

4. La stratégie numérique pour la Wallonie développée dans le cadre du plan Marshall 4.0 .
<https://www.digitalwallonia.be/>

5. Sous license GNU LGPL 3.0 et CC-BY-SA 3.0

différents types de capteurs sont illustrés en bleu dans la figure 2.2 tandis que les actionneurs le sont en orange. En vert, se trouvent quelques fonctionnalités supplémentaires : un trou central pouvant accueillir un marqueur pour faire des dessins, des attaches pour Lego pour décorer le robot, le dongle USB et le module Wireless. Les capteurs de Thymio lui permettent de percevoir les obstacles, les sons, les inclinaisons, les chocs, les infra-rouges, la température, les interactions avec ses boutons capacitifs et enfin, l'état de sa batterie. Ses actionneurs lui permettent d'émettre des sons, de la lumière et de se mouvoir. La présence de ces LEDs est un des plus gros avantages de Thymio II par rapport aux autres robots éducatifs. En effet, ils aident à comprendre le fonctionnement des capteurs en s'allumant lorsqu'un capteur de proximité détecte un obstacle. C'est très intuitif pour les enfants.

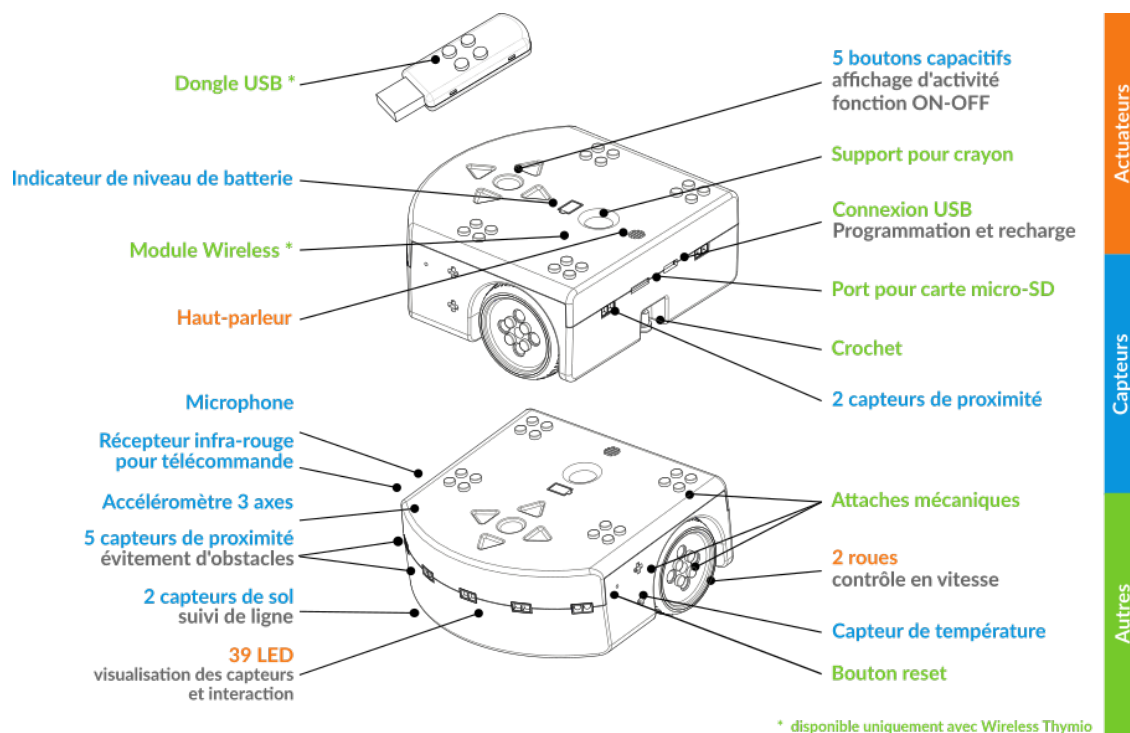


FIGURE 2.2 – Composition d'un Wireless Thymio
(Source : <https://www.thymio.org/>)

Le logiciel

Les robots Thymio II sont programmables de diverses manières à l'aide du logiciel Aseba⁶. Ce logiciel facilite la gestion des robots possédant plusieurs processeurs pour les chercheurs. Après un essai concluant au Festival de Robotique 2008, il a été choisi pour les Thymio. Aujourd'hui, le logiciel permet de programmer les Thymio de trois manières différentes :

- En programmation textuelle, avec le langage Aseba qui permet d'utiliser toutes les fonctions du robot, c'est-à-dire que toutes les valeurs des capteurs peuvent être utilisées, cela permet donc beaucoup de précision.

6. Développé par Dr. S. Magnenat lors de sa thèse à l'EPFL puis à l'EPF Zürich.

```

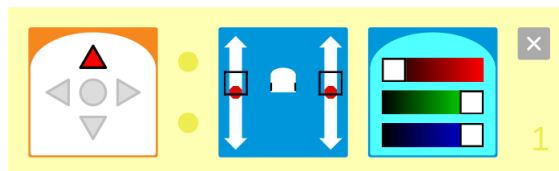
onevent buttons
  when button.forward == 1 do
    motor.left.target = 150
    motor.right.target = 150
    call leds.top(0,32,32)
    emit pair_run 0
  end

```

- En programmation par blocs, soit avec Blockly⁷, soit avec Scratch⁸. Ces deux outils permettent également d'utiliser toutes les valeurs des capteurs et donc de programmer de manière précise. Blockly est directement intégré à Aseba Studio tandis que Scratch est disponible sous forme d'un plugin. La principale différence entre Blockly et Scratch est que Blockly est basé sur la programmation événementielle alors que Scratch a plutôt une approche séquentielle (cf. section 1.2).



- En programmation visuelle, grâce au langage VPL⁹ qui permet de programmer le robot en associant des blocs d'action à des blocs d'événement. Même s'il existe un mode VPL avancé, ce langage ne permet pas d'utiliser les capacités du robot au maximum car les valeurs des capteurs ne peuvent pas être définies à la main, cela manque donc de précision.



Pour cette séquence de cours, nous avons opté pour la programmation visuelle. En effet, elle a l'avantage d'être très intuitive et ne contraint pas les enfants à devoir lire énormément de blocs, ce qui permet de gagner du temps pour ceux qui ne seraient pas encore tout à fait à l'aise avec la lecture. Le langage VPL fonctionne sur base de pictogrammes (figure 2.3) qu'il faut assembler pour créer des instructions. Une instruction est composée d'un ou plusieurs bloc(s) d'événement (en orange) et d'un ou plusieurs bloc(s) d'action (en bleu). Les pictogrammes sont divisés en deux séries : il y a ceux de base, formant ce qu'on appelle le VPL simple, et ceux qui forment le VPL avancé, permettant une plus grande précision des seuils de détection, l'utilisation de modes et de timers, et l'emploi de l'accéléromètre.

En plus d'être programmable, Thymio contient déjà des programmes pré-enregistrés qui sont appelés comportements. Ils sont au nombre de six, sont chacun associés à une couleur et ont chacun des caractéristiques différentes.

7. Une bibliothèque JavaScript développée par Google et permettant la création d'environnements utilisant un langage de programmation graphique.

8. Un langage de programmation graphique développé par le MIT.

9. Le Visual Programming Language développé par Dr. J. Shin et Dr. S. Magnenat.

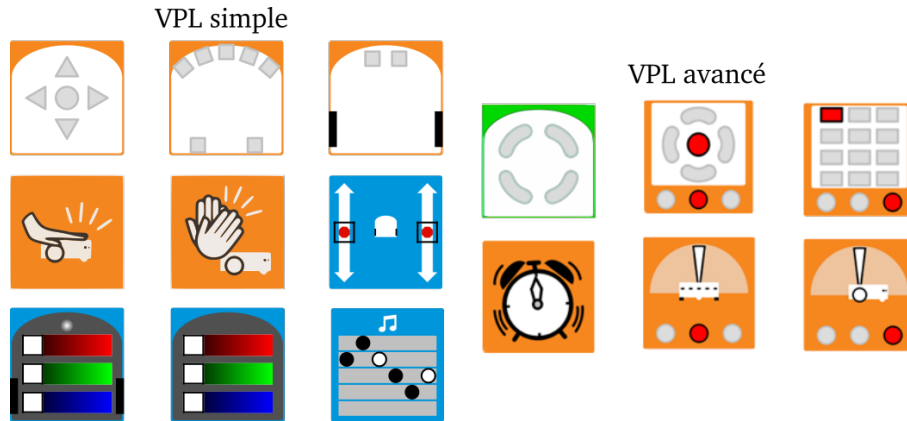


FIGURE 2.3 – Les différents pictogrammes du langage VPL

- Le vert a pour caractéristique d'être amical : lorsqu'il perçoit un obstacle, il se maintient toujours à une même distance de cet obstacle et le suit donc.
- Le jaune est comparé à un aventurier, car il avance en ligne droite jusqu'à percevoir un obstacle qu'il évite et poursuit ensuite sa route.
- Le rouge est le comportement le plus peureux. Il fuit dans la direction opposée à l'obstacle qu'il détecte et "crie" lorsqu'il se retrouve bloqué entre deux obstacles ou lorsqu'il est en chute libre.
- Le rose est obéissant. Il se laisse diriger avec les flèches présentes sur le robot ou à l'aide d'une télécommande.
- Le bleu est attentif car il réagit aux sons. Il est un peu délaissé lors de cette séquence car la perception du bruit dans une salle de classe n'est pas chose aisée.
- Et enfin, le cyan est dit curieux, car il suit les pistes de bandes noires qu'on lui laisse au sol. Il est un peu moins utilisé puisqu'il nécessite un matériel particulier.

2.2 Présentation de la séquence

Cette section a pour but de présenter la séquence de cours telle qu'elle avait été pensée à la base. Il s'agit de 12 séances de cours de 50 minutes chacune. Elles sont présentées de manière synthétique dans le tableau de la première sous-section et plus en détails, avec une description de chaque activité prévue, dans la seconde.

Certaines de ces activités sont tirées ou inspirées du guide pédagogique *1, 2, 3... Codez!* [15] ou des Challenge Activities for Thymio¹⁰.

Tableau récapitulatif de la séquence

Le tableau suivant reprend toutes les activités de la séquence ainsi que les objectifs qu'elles poursuivent et le matériel qui leur est nécessaire. En jaune, ce sont les activités qui se font sur papier ; en vert, ce sont les activités qui se font avec le robot ; et enfin, en bleu, les activités qui nécessitent un ordinateur. Toutes les activités en bleu travaillent le pair programming, ce n'est donc pas indiqué dans le tableau.

10. Qui peuvent être trouvées directement sur le site de Thymio (<https://www.thymio.org/fr/creations/>)

Séances	Sigle	Activités	Objectifs, notions et compétences	Matériel
Introduction au robot	S1.1	Qu'est-ce qu'un robot ?	Concept de robot	-
	S1.2	Prise en main de Thymio	Observation de ce que Thymio peut faire	Thymio
	S1.3	Explication des capteurs, actionneurs, ...	Notions de capteur, actionneur, de boucle sensori-motrice, ...	-
Observer les comportements de Thymio	S2.1	Comprendre un des comportements par groupe	Test systématique des capteurs pour tirer des conclusions	Thymio
	S2.2	Un élève joue le rôle du robot	Importance des capteurs	-
	S2.3	Redéfinition de ce qu'est un robot	Elaboration d'une définition Notion de robot	-
Formalisation des comportements de Thymio	S3.1	Formalisation d'un comportement		Sets de pictogrammes
	S3.2	Exposer la formalisation d'autres comportements	Formalisation	Copies des annexes A et B Copie des formalisations des autres comportements
	S3.3	Nommer les comportements	Empathie	-
Utilisation des différents comportements	S4.1	Sortir d'un labyrinthe avec tous les comportements	Décortiquer un problème Trouver différentes approches pour résoudre un même problème	Thymio Bandes noires Kaplas Thymio Petites balles
	S4.2	Différents challenges		Thymio Bandes noires Obstacles en carton
Introduction à la programmation VPL	S5.1	Challenge irrésolvable avec les comportements pré-enregistrés	Résolution de problème Découpage en sous-problèmes	Thymio Bandes noires Obstacles en carton
	S5.2	Présentation d'Aseba Studio	Familiarisation avec les outils utilisés	Copie des pictogramme
	S5.3	Petites tâches simples à réaliser	Prise en main du langage VPL	Ordinateurs Thymio

Séances	Sigle	Activités	Objectifs, notions et compétences	Matériel
Programmation simple de Thymio	S6.1	Enchaînement d'instructions	Prise en main du langage VPL Apprentissage de la programmation événementielle	Copies du cours avec les énoncés d'exercices
	S6.2	Vrai ou faux		
	S6.3	Débugging		
	S6.4	Piano coloré	Programmation événementielle Décortiquer un problème Comprendre comment donner des instructions à un robot	Ordinateur Thymio
	S6.5	Reproduire le comportement rouge		
Programmes un peu plus complexes	S7.1	Reproduire le comportement cyan	Traduire les pictogrammes en français	Ordinateurs Thymio Bandes noires Ordinateur Thymio Copies du cours avec l'énoncé de l'exercice
	S7.2	Prédiction sur des programmes		
Résolution de notre problématique de départ	S8.1	Programmer le déménéageur	Programmation événementielle Décortiquer un problème Comprendre comment donner des instructions à un robot	
		Le robot fait quelque chose et ils doivent trouver le programme correspondant		
	S8.2	Traduire un comportement en français		
Quiz final	S9.1	Programmer les Thymio en mode buzzer	Programmation événementielle Décortiquer un problème Comprendre comment donner des instructions à un robot	Ordinateur Thymio
	S9.2	Quiz	Evaluation des connaissances basiques liées au robot et à la programmation visuelle	Thymio

Séances	Sigle	Activités	Objectifs, notions et compétences	Matériel
Programmation VPL avancée (intro)	S10.1	Présentation des nouveautés	Prise en main du langage VPL avancé	Copie des pictogrammes
	S10.2	Thymio change de couleur (timer et modes)	Programmation événementielle	Ordinateurs Thymio
	S10.3	Thymio freine progressivement	Décortiquer un problème	Ordinateurs Thymio Obstacles
Programmation VPL avancée (suite)	S11.1	Le déménageur		Ordinateurs Thymio Bandes noires
	S11.2	Une chorégraphie		Obstacles Ordinateurs Thymio Bandes noires
Thymio et le code de la route	S12.1	Programmer le feu de signalisation		
	S12.2	Programmer les alertes	Programmation événementielle	Ordinateur
	S12.3	Faire que Thymio poursuive l'autre	S'atteler à un projet plus ambitieux	Thymio

Tableau 2.1 – Récapitulatif de toutes les activités pensées pour la séquence sur la robotique

2.2.1 Séance 1 - Introduction au robot

Cette séance de cours a pour but de faire une introduction aux robots et aux concepts qui y sont liés. Trop souvent, les enfants pensent que les robots sont uniquement des humanoïdes que l'on essaie de faire ressembler au niveau émotionnel, intellectuel et physique à un être humain. Ils ne se rendent pas compte qu'ils en côtoient régulièrement. C'est donc l'occasion de déconstruire leurs idées reçues sur le sujet et de repartir sur de bonnes bases.

S1.1 - Qu'est-ce qu'un robot ? [15 min]

Il s'agit d'une activité de brainstorming qui vise à connaître la perception qu'ont les enfants d'un robot. Aucun matériel n'est nécessaire puisqu'il s'agit d'une discussion. Plusieurs questions sont posées afin de s'assurer de couvrir les différents éléments d'intérêt. Ce sont des questions du type "À quoi ça sert?", "À quoi ça ressemble?", "Comment ça marche?" et "Pouvez-vous donner quelques exemples?". Ils citent des exemples. Les informations importantes sont reprises au tableau pour pouvoir y revenir plus tard. À noter que, si l'occasion se présente, cela peut permettre de faire la distinction entre un automate et un robot.

S1.2 - Prise en main de Thymio [15 min]

Par groupe de deux ou plus, si le nombre de Thymio ne le permet pas, ils reçoivent un robot et ils ont 10 minutes pour l'appivoiser en discutant entre eux : comprendre comment l'allumer et comment sélectionner les différents comportements pré-enregistrés. Ensuite, ils mettent leurs découvertes en commun avec la classe. Le but est qu'ils remarquent que Thymio est capable de se mouvoir, d'émettre des sons et de changer de couleur.

S1.3 - Explications des capteurs, actionneurs, ... [20 min]

Après la discussion concernant la prise en main de Thymio, les groupes remarquent que le robot réagit à son environnement. Certains binômes auront remarqué que parfois le robot évite les obstacles ou bien qu'il suit leur main. Vient donc la question de savoir comment il fait. Après un temps de parole, en les aiguillant un peu, la notion de capteur est introduite, puis celle d'actionneur. Un parallèle peut donc être fait entre les capteurs et les yeux/oreilles, mais également entre les actionneurs et les bras/jambes ou plutôt les muscles.

Étant donné que Thymio est facile à démonter, il peut être ouvert pour montrer les différents éléments qui se trouvent à l'intérieur. C'est le moment de présenter les capteurs, les actionneurs et l'ordinateur qui permettent que tout cela fonctionne. Finalement, les enfants sont invités à rédiger une brève description du robot.

2.2.2 Séance 2 - Observer les comportements de Thymio

Cette séance porte sur les différents comportements pré-enregistrés de Thymio. Les enfants observent les différents comportements et se mettent à la place du robot pour réaliser l'importance des capteurs et de leur fonctionnement.

S2.1 - Comprendre un des comportements par groupe [20 min]

Par petits groupes, afin de favoriser les échanges et l'entraide, ils reçoivent un Thymio avec un certain comportement. Les comportements pré-sélectionnés sont le rouge, le jaune, le vert ou le rose. Les deux bleus ne sont pas utilisés car les enfants n'ont pas les bandes noires nécessaires au comportement cyan et le bruit ambiant rend l'utilisation du bleu compliquée. L'attention est portée sur le test systématique de tous les capteurs mentionnés précédemment. L'activité est suivie d'une mise en commun de leurs observations.

S2.2 - Un élève joue le rôle du robot [15 min]

Un enfant a les yeux bandés, ses mains feront office de capteurs. Il doit se diriger de lui-même vers un autre endroit de la classe, ou suivre un autre élève qui peut accélérer ou s'arrêter à tout moment. Dans un premier temps, le reste de la classe observe. Ensuite, l'activité peut être faite en binôme. L'important étant que les enfants réalisent l'utilité et le fonctionnement des capteurs.

S2.3 - Redéfinition de ce qu'est un robot [15 min]

En repartant des idées énoncées lors du brainstorming du début de séance, les enfants font le tri entre le vrai et le faux. Ils peuvent également ajouter de nouveaux éléments. Finalement, chaque enfant est invité à rédiger une définition d'un robot en prenant soin d'y inclure les nouveaux concepts vus lors de la séance, en utilisant un vocabulaire précis et adapté. Chacun doit parvenir à inclure l'idée de programme, de capteurs et d'actionneurs dans sa définition. La définition idéale que l'on souhaite obtenir est celle présentée au point 2.1.2, on peut donc aiguiller les enfants jusqu'à avoir une définition suffisamment proche.

2.2.3 Séance 3 - Formalisation des comportements de Thymio

Le but de cette séance est de formaliser les comportements observés à la séance précédente. Cette séance fait également appel à leur empathie pour pouvoir nommer les différents comportements de Thymio.

À partir de cette séance-ci, les enfants commencent à travailler la compétence de saisie de l'information présente dans les socles de compétences [14]. Celle-ci consiste à utiliser d'autres codes ou langages en utilisant les pictogrammes de VPL pour programmer ou décrire un comportement de Thymio.

S3.1 - Formalisation d'un comportement [25 min]

La séance débute par une présentation des divers pictogrammes de VPL qui sont nécessaires à la formalisation des comportements jaune et rouge. La première partie de l'activité se fait tous ensemble. Chaque enfant reçoit une page (cf. annexe A) et un set de petits pictogrammes à adapter (cf. figure 2.4). La distinction entre les événements (en orange) et les actions (en bleu) est bien marquée à la fois sur la feuille reçue et sur les pictogrammes.

Le but est donc de reconstituer le comportement jaune en associant un événement à une action et en coloriant sur les pictogrammes les capteurs utilisés ou en indiquant le

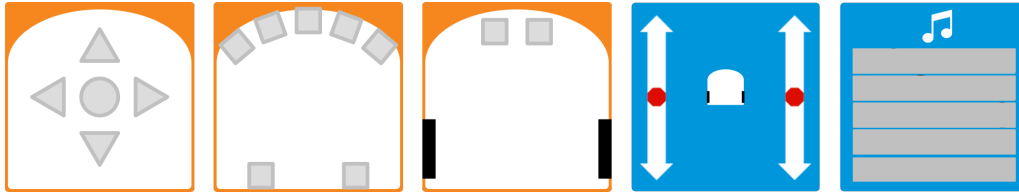


FIGURE 2.4 – Contenu du set de pictogrammes à adapter reçu par les enfants

positionnement des moteurs. Les enfants ont toujours le robot à disposition pour tester les capteurs un à un. Le résultat attendu est disponible dans l'annexe A.

La deuxième partie de l'activité consiste à faire la même chose avec le comportement rouge, mais seul cette fois. Ils reçoivent donc de nouveau une page qui donne une indication du nombre d'instructions attendues et un set de pictogrammes. À la fin de l'activité, il y a une mise en commun pour vérifier que chacun a bien compris.

Lors de cette activité, les enfants sont amenés à morceler un problème en petits sous-problèmes qui sont ici des instructions. Ils ne peuvent plus se contenter de dire "Thymio évite les obstacles.", ils doivent comprendre comment le robot fait pour les éviter.

S3.2 - Exposer la formalisation d'autres comportements [15 min]

Pour que les enfants perçoivent bien le fonctionnement des différents comportements, leur formalisation est exposée. Chaque formalisation est présentée sans dire de quel comportement il s'agit. Pour le déterminer, les enfants sont invités à décrire ce que font les différentes instructions.

S3.3 - Nommer les comportements [10 min]

Les enfants sont conviés à trouver des noms pour les différents comportements pré-enregistrés de Thymio. Ils doivent les nommer par des traits de caractère. De cette manière on s'attend à avoir des noms tels que peureux (rouge), aventurier (jaune), amical (vert), attentif (bleu), obéissant (rose) et curieux (cyan).

2.2.4 Séance 4 - Utilisation des différents comportements

Cette séance a pour but de laisser les enfants mettre en pratique ce qu'ils ont appris des différents comportements à l'aide de petits challenges.

S4.1 - Sortir d'un labyrinthe avec tous les comportements [30 min]

Par groupe de quatre, ils sont assignés à un labyrinthe délimité par des blocs en bois posés sur le sol. Ils possèdent également quelques bandes de papier noir. Leur but est de faire sortir le robot du labyrinthe avec tous les comportements pré-enregistrés de Thymio (selon le niveau de bruit de la classe, le comportement bleu foncé devra peut-être être abandonné). À la fin du temps imparti, ils mettent en commun les difficultés ou facilités qu'ils ont rencontrées. Cela permet de se rendre compte que certains comportements sont plus propices pour réaliser certaines tâches.

S4.2 - Différents challenges [20 min]

Pour finaliser la prise en main de Thymio, les enfants vont réaliser quelques petits challenges¹¹. Le but est de trouver le(s) comportement(s) le(s) plus adapté(s) pour résoudre un problème donné.

- La chenille : les enfants doivent se concerter pour que tous les Thymio forment une chenille capable de se promener de manière autonome dans la classe. Comme cette activité concerne toute la classe, il faut que les enfants s'écoutent et puissent donner leur point de vue de manière à se coordonner.
- Les tirs au but : par groupe de deux, les enfants ont une balle et un goal. Ils doivent trouver le comportement de Thymio le plus adapté à une séance de tirs au but.

2.2.5 Séance 5 - Introduction à la programmation VPL

Cette séance montre les limites des comportements pré-enregistrés pour pouvoir introduire ensuite la programmation VPL.

S5.1 - Challenge irrésolvable avec les comportements pré-enregistrés [10 min]

Les enfants font face au problème du robot déménageur, qui est en fait irrésolvable avec les comportements pré-enregistrés du robot. Ce problème consiste à placer le robot à l'intérieur d'un rectangle étroit dessiné au sol et délimité par des bandes noires. Plusieurs obstacles sont disposés à l'intérieur de ce rectangle et le robot doit les pousser vers l'extérieur.

Pour pouvoir accomplir sa mission, le robot doit avancer dans la direction où il perçoit un obstacle et changer de direction quand il atteint une ligne noire au sol. S'il n'y a ni ligne noire, ni obstacle, le robot avance en ligne droite jusqu'à en trouver. Ceci n'est donc pas possible à faire avec les comportements pré-enregistrés puisque la plupart des comportements s'arrêtent quand ils perçoivent une ligne noire au sol. Le seul comportement suivant les lignes noires est le comportement cyan. Malheureusement, en l'absence de ces dernières, il tourne sur lui-même et ne trouverait donc pas les obstacles à déplacer.

Les enfants ont quelques instants pour discuter en petits groupes et trouver comment résoudre le problème avec les comportements pré-enregistrés. Si certains n'arrivent pas directement à la conclusion que c'est impossible, d'autres ont la possibilité de leur expliquer pourquoi ça ne fonctionne pas. Une fois qu'il est établi que ce n'est pas possible, les enfants doivent répondre à la question de ce que devrait faire le robot pour que cela fonctionne. L'activité se termine par une insistance sur le concept de programmation.

S5.2 - Présentation d'Aseba Studio [20 min]

L'activité débute par un rappel des pictogrammes déjà rencontrés lors de l'étape de formalisation des comportements. À cela s'ajoutent les derniers pictogrammes utiles dans la version de base de VPL (cf. figure 2.3). Ensuite, présentation et explication de l'interface.

11. Tirés des Challenge Activities for Thymio qui peuvent être trouvés ici : <https://www.thymio.org/fr/creations/>

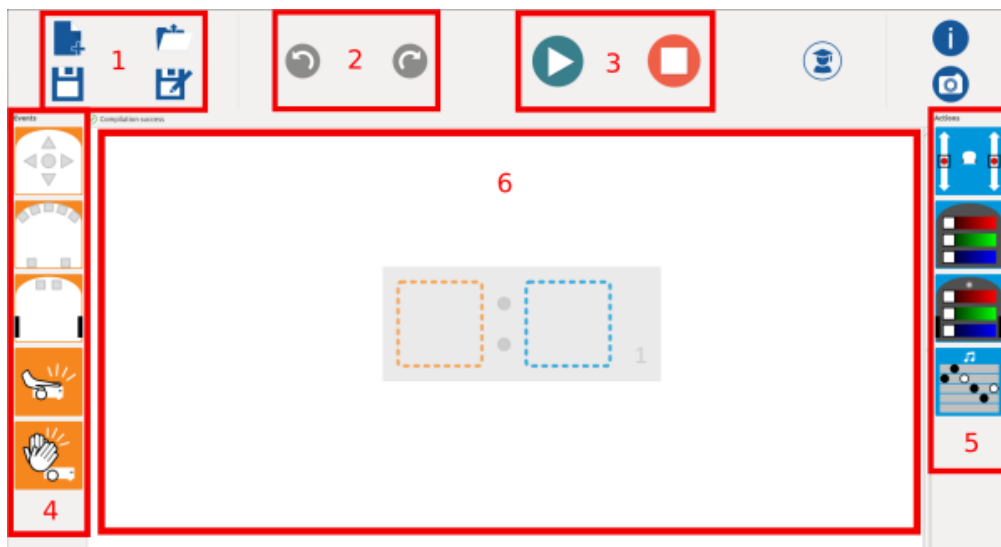


FIGURE 2.5 – Interface d’Aseba VPL

1. Zone qui gère l’enregistrement, la création ou l’importation des programmes VPL.
2. Zone qui gère le retour en arrière ou en avant en cas d’erreur.
3. Zone qui gère l’exécution ou l’arrêt du programme. Le triangle devient vert lorsque le programme peut être exécuté.
4. Différents pictogrammes d’événement.
5. Différents pictogrammes d’action.
6. Zone de programmation. En glissant à chaque fois un événement et au moins une action, une instruction va se créer.

Démonstration de l’exécution d’un programme simple. L’attention est portée sur le changement de couleur de l’instruction en train d’être exécutée (cf. figure 2.6) puisque le feedback visuel réduit le taux d’erreur dans les exercices [16]. C’est également l’occasion



FIGURE 2.6 – Feedback visuel offert par la programmation VPL (avec la flèche jaune, l’instruction en train d’être exécutée)

de mentionner que Thymio ne comprend pas les pictogrammes et que ces derniers sont traduits dans un langage de programmation qui, lui, sera compris par la machine.

S5.3 - Petites tâches simples à réaliser [20 min]

Pour leurs premiers pas en programmation, les enfants doivent réaliser des petites tâches simples pour tester leur compréhension de l'interface et des nouveaux pictogrammes. Il s'agit d'une simple instruction ou parfois de deux maximum.

- Thymio devient rouge quand on lui tape (gentiment !) dessus.
- Thymio avance quand on appuie sur la flèche avant et il s'arrête quand il perçoit un obstacle droit devant lui.
- Thymio tourne sur lui-même, vers la gauche, plus ou moins vite quand on appuie sur les flèches de devant ou de derrière.
- Thymio émet un son quand on appuie sur le bouton du milieu.

2.2.6 Séance 6 - Programmation simple de Thymio

Cette première séance de programmation permet aux enfants de se familiariser avec la formation d'instructions par de petits exercices simples. À partir de cette séance, les enfants travaillent en pair programming.

S6.1 - Enchaînement d'instructions [5 min]

Premier exercice d'une série un peu plus théorique ayant pour but d'asseoir leur compréhension du langage VPL. Il y a plusieurs phrases en français décrivant une action et deux instructions en VPL (cf. figure 2.7). Les enfants doivent choisir l'instruction qui correspond à la phrase en français. Ils sont installés à deux par machine et peuvent donc s'entraider.

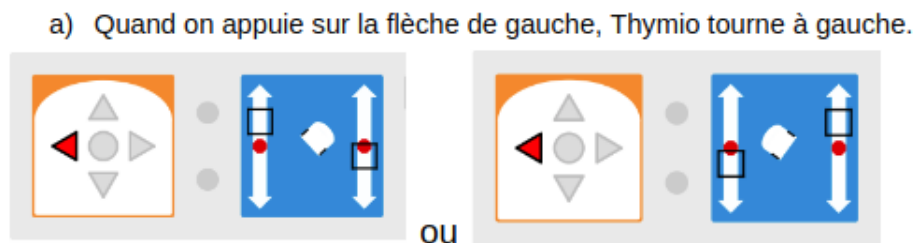


FIGURE 2.7 – Exemple d'exercice d'enchaînement d'instructions

S6.2 - Vrai ou faux [5 min]

Les enfants reçoivent plusieurs petits codes de deux instructions accompagnés d'une phrase en français décrivant un comportement. Ils doivent déterminer si la description est correcte ou non, et ils doivent justifier leur réponse (cf. figure 2.8). De nouveau, ils sont à deux par machine et peuvent s'entraider.

S6.3 - Debugging [10 min]

Les enfants reçoivent un programme plus long, toujours accompagné d'un texte descriptif d'un comportement (cf. annexe B). Ils doivent y retrouver trois erreurs et les expliquer. Les

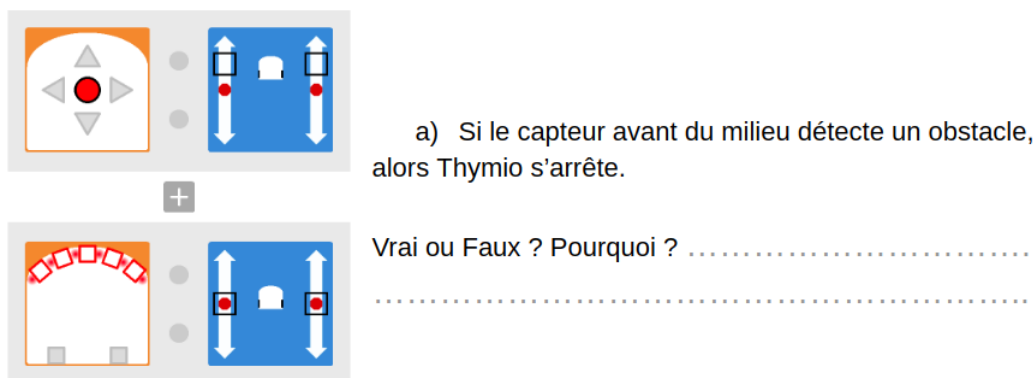


FIGURE 2.8 – Exemple d'exercice de vrai ou faux

deux premières erreurs ne sont pas difficiles à trouver, il s'agit de problèmes d'utilisation des pictogrammes. Par contre, la troisième mène à un point très intéressant. On ne s'attend pas à ce que tous les enfants la trouvent ce qui laissera une opportunité d'attirer leur attention dessus. Les enfants sont informés que le programme tel qu'il est ne peut fonctionner et ils doivent trouver pourquoi. Le problème vient de la redondance des instructions, car si le robot doit produire deux actions différentes (ici, reculer et s'arrêter) en même temps, il ne sait pas quoi faire et le programme ne s'exécute donc pas.

S6.4 - Piano coloré [15 min]

Les enfants ont pour mission de transformer Thymio en un piano qui fait des couleurs. Pour cela, ils doivent associer chacun des capteurs avant du robot à un son et à une couleur. Ils peuvent ensuite tester leur nouvel instrument de musique. S'ils sont rapides, ils peuvent ajouter des sons et couleurs pour tous les autres capteurs.

S6.5 - Reproduire le comportement rouge [15 min]

La dernière mission de cette séance consiste à recréer une version simplifiée du comportement rouge, car le VPL simple ne permet pas de le reproduire entièrement. Pour arriver à cela, les enfants doivent déconstruire le comportement comme ils l'ont fait lors de sa formalisation. Ils sont aiguillés pour réaliser ce programme progressivement : faire en sorte que Thymio se déplace dans la direction opposée à son capteur enclenché, pour les capteurs à l'avant, puis pour les capteurs à l'arrière, et ensuite, programmer le cri quand il est bloqué. Ils peuvent également programmer la sécurité qui empêche Thymio de tomber de la table.

2.2.7 Séance 7 - Programmes un peu plus compliqués

Maintenant que les enfants sont plus à l'aise avec l'interface et le langage VPL, ils peuvent travailler sur des programmes qui demandent un peu plus de réflexion.

S7.1 - Reproduire le comportement cyan [25 min]

Dans la même optique que la dernière activité de la séance précédente (activité S6.5), les enfants vont devoir reproduire un des comportements pré-enregistrés de Thymio : le cyan, c'est-à-dire, celui qui suit les bandes noires. Pour cette activité, les enfants testent par eux-mêmes avec le robot et, le cas échéant, cherchent à comprendre pourquoi il ne fait pas exactement ce qu'ils attendent de lui. Le robot doit avancer en ligne droite jusqu'à ce qu'il trouve une ligne noire au sol et doit la suivre malgré les virages.

S7.2 - Prédiction sur des programmes [25 min]

Plusieurs codes sont présentés aux enfants et ils doivent rédiger avec soin une phrase décrivant le comportement représenté par l'instruction (cf. figure 2.9). Lorsque c'est fait, ils programment eux-mêmes le code fourni et vérifient si leurs prédictions sont correctes ou non. Si elles ne le sont pas, ils en rediscutent en binôme et reformulent leur prédiction.

Détermine ce que fait le code, puis programme-le pour vérifier ta réponse.



Que fait Thymio ?

.....

FIGURE 2.9 – Exemple d'exercice de prédiction

2.2.8 Séance 8 - Résolution de notre problématique de départ

Dans cette séance, les enfants sont amenés à résoudre un problème, celui du robot déménageur vu plus tôt. Ils affinent toujours un peu plus leur maîtrise de la programmation en VPL.

S8.1 - Programmer le déménageur [30 min]

Pour ne pas laisser un problème sans solution, les enfants vont programmer le robot afin qu'ils puissent résoudre le problème du déménageur vu lors de la séance 5 (section 2.2.5). Ils sont censés maîtriser tous les outils nécessaires pour décomposer le problème en sous-problèmes et pour les traduire avec les instructions. Ce problème peut être résumé comme suit : quand il perçoit un objet, Thymio avance tout droit et s'il perçoit une ligne noire au sol, alors il change un peu de direction afin de rester dans le rectangle tracé au sol.

S8.2 - Le robot fait quelque chose et ils doivent trouver le programme correspondant [20 min]

Après avoir traduit le français en code, le code en français et le code en actions effectuées par le robot, il ne manque plus qu'à traduire les actions effectuées par le robot en code. Pour cela, on leur montre la vidéo proposée dans le MOOC sur Thymio¹². Il s'agit d'une vidéo montrant six Thymio ayant chacun des comportements distincts. Les enfants reçoivent différents programmes et ils doivent associer chaque Thymio vu dans la vidéo à un des programmes qu'ils ont reçus.

2.2.9 Séance 9 - Quiz final

À l'issue de cette partie concernant le VPL simple, les enfants peuvent montrer ce qu'ils ont compris et retenu des séances précédentes.

S9.1 - Programmer les Thymio en mode buzzer [20 min]

Avant de procéder au quiz final, les enfants créent leur propre buzzer par binôme toujours. Pour cela, ils doivent faire en sorte que le robot s'allume dans quatre couleurs différentes : rose, jaune, vert et bleu. Ils sont libres de décider s'ils veulent utiliser les flèches ou les capteurs pour le faire.

S9.2 - Quiz [30 min]

Les enfants prennent part à un quiz de 12 questions à choix multiples. C'est là que vont servir les buzzers créés juste avant. À chaque réponse est associée une couleur du robot, les enfants indiquent leur choix en colorant le robot. Cela a deux avantages : c'est visible de loin et ça évite le cri des réponses dans la salle de classe.

Le but de ce quiz est de permettre d'évaluer la compréhension des enfants, de cibler les informations qui ont pu leur échapper et d'apporter certains éclaircissements. Le quiz est constitué de questions de programmation, de questions sur les concepts, de questions de mémoire, etc.

Voici quelques exemples de questions

1. Qu'est-ce qui permet à Thymio de percevoir ce qui l'entoure ?
2. Qu'est-ce qu'un robot ?
3. Combien de capteurs de proximité Thymio possède-t-il ?
4. Que fait ce programme ?
5. Quel type de programmation avez-vous expérimenté avec Thymio ?
6. Que représentent les pictogrammes orange ?
7. Quelle est l'instruction qui s'exécute ?
8. Comment appelle-t-on le fait de programmer à deux devant un ordinateur ?

12. <https://www.fun-mooc.fr/courses/course-v1:inria+41017+session01/info>

2.2.10 Séance 10 - Programmation VPL avancée - Introduction

Maintenant que les enfants ont bien l'outil VPL en main, ils sont capables de passer au niveau supérieur : le VPL avancé. De nouveaux pictogrammes sont introduits (cf. figure 2.3), l'accéléromètre est maintenant disponible et les notions de mode et de timer sont présentées. Tout cela permet de faire des programmes encore plus précis avec le robot.

S10.1 - Présentation des nouveautés [15 min]

La séance commence par une présentation des nouveaux pictogrammes qui peuvent être utilisés. Une explication est donnée pour les différentes "facettes" de certains pictogrammes, par exemple, les différents axes d'inclinaison liés à l'accéléromètre. Les modes sont présentés à l'aide d'une machine à état, le fonctionnement des timers est expliqué et on introduit la distance de perception pour les capteurs. Les enfants sont conviés à réaliser quelques petits exercices théoriques dans le même style que ceux présentés à la séance 6 (section 2.2.6).

S10.2 - Thymio change de couleur (timers et modes) [15 min]

Les enfants programment le robot pour qu'il change de couleur après quelques secondes. Grâce à cela, ils découvrent les timers et les modes du VPL avancé. Ils travaillent toujours en pair programming de manière à pouvoir s'entraider.

S10.3 - Thymio freine progressivement [20 min]

Pour se familiariser avec les distances perçues par les capteurs, les enfants programment Thymio pour qu'il freine progressivement en approchant d'un obstacle jusqu'à s'arrêter. Pour qu'ils aient plus de facilité à déboguer leur programme en observant le robot, il leur est conseillé de colorer le robot différemment à chaque décélération. Cette activité n'est pas simple, il y a quelques paramètres à prendre en compte, tels que la vitesse de base, pour que le robot arrive à s'arrêter à temps.

2.2.11 Séance 11 - Programmation VPL avancée - Suite

Cette séance fait appel à toutes les connaissances qu'ils ont acquis au fur et à mesure des activités. Elle fait également appel à leur créativité car les activités ne sont (presque) pas guidées, ils peuvent donc résoudre les problèmes comme bon leur semble.

S11.1 - Le déménageur [25 min]

On reprend de nouveau le problème du déménageur. Cette fois, on ne veut plus qu'il suive les bords et qu'il pousse les obstacles hors du rectangle, on souhaite qu'il cherche les obstacles à l'intérieur de l'espace afin de pouvoir les pousser hors des limites.

S11.2 - Une chorégraphie [25 min]

La principale consigne de cette activité est de s'amuser ! Il faut faire danser Thymio en utilisant les mouvements, les couleurs, les timers et les modes. Ils peuvent s'appuyer sur une musique de leur choix s'ils veulent que le robot danse en rythme.

2.2.12 Séance 12 - Thymio et le code de la route

La toute dernière séance repose sur un projet d'envergure. Un Thymio fait office de feu de signalisation, il change de couleur et lance un alerte quand il voit passer un obstacle lorsqu'il est orange et se lance dans une course poursuite s'il voit passer un obstacle lorsqu'il est rouge. Les enfants doivent décomposer ce comportement afin de pouvoir le programmer.

S12.1 - Programmer le feu de signalisation [15 min]

Les enfants commencent en programmant le feu à proprement parler. Thymio est vert pendant 4 secondes, orange pendant 2 secondes et rouge pendant 3 secondes et puis il redevient vert.

S12.2 - Programmer les alertes [15 min]

Ensuite, les enfants programment les alertes. Si un obstacle passe devant le Thymio lorsqu'il est orange, il sonne pour donner un avertissement. Par contre, si l'obstacle passe devant le Thymio lorsqu'il est rouge, il sonne et se met à clignoter en rouge et bleu.

S12.3 - Faire que Thymio poursuive l'autre [15 min]

Et pour finaliser ce beau projet d'envergure, il faut faire en sorte que lorsqu'un obstacle passe devant le Thymio lorsqu'il est rouge, ce dernier se mette en marche et parte à la poursuite de l'obstacle.

2.3 Retours et améliorations

Cette section contient les retours de ces séances de cours données dans une classe de primaire. L'école a alloué 8h de cours à cette séquence, il a donc fallu faire des choix pour sélectionner les activités à garder ou non. Ces choix sont expliqués dans la sous-section suivante. Ensuite, le test des séances en classe a permis de se rendre compte d'éventuels problèmes, la deuxième sous-section explique donc les améliorations possibles. Enfin, une analyse générale est réalisée pour chercher les causes des soucis survenus.

2.3.1 Déroulement des séances

La séquence de cours a été donnée à l'école des Bruyères¹³ à Louvain-la-Neuve dans une classe de 5ème primaire pour 21 élèves. Huit séances de 50 min ont été prévues dans la classe et elles ont été réalisées par bloc de deux en l'espace de 4 semaines. Le tableau 2.2 reprend les activités qui ont été sélectionnées pour être réalisées en classe. Nous expliquons les raisons de ces choix dans la suite de cette section.

13. <https://www.bruyeres.be/>

Séances	Activités		Compétences et objectifs
	Sigle	Nom	
1	<p>S1.1 Qu'est-ce qu'un robot ?</p> <p>S1.2 Prise en main de Thymio</p> <p>S1.3 Explications des capteurs, actionneurs, ...</p> <p>S2.1 Comprendre un des comportements par groupe</p> <p>S2.3 Rédéfinition de ce qu'est un robot</p> <p>S3.1a Formalisation d'un comportement tous ensemble</p>	<p>Introduction au robot</p> <p>Déconstruction des idées reçues sur les robots</p> <p>Définition d'un robot</p> <p>Notions liées au robot</p> <p>Première manipulation de Thymio</p> <p>Appropriement des comportements</p> <p>Formalisation</p>	
2	<p>S3.1b Formalisation d'un comportement seul</p> <p>S3.2 Exposer la formalisation des autres comportements</p> <p>S3.3 Nommer les comportements</p> <p>S4.1 Sortir d'un labyrinthe avec tous les comportements</p> <p>S4.2 Différents challenges</p> <p>S5.2 Présentation d'Aseba Studio</p>	<p>Formalisation et apprivoisement des comportements de Thymio</p> <p>Formalisation</p> <p>Empathie</p> <p>Utilisation d'autres codes (VPL) pour décrire le comportement de Thymio</p> <p>Morceler un problème pour pouvoir le résoudre</p>	
3	<p>S6.1 Enchaînement d'instructions</p> <p>S6.2 Vrai ou faux</p> <p>S6.3 Débugging</p> <p>S5.3 Petites tâches simples à réaliser</p> <p>S6.4 Le piano coloré</p> <p>S6.5 Reproduire le comportement rouge</p> <p>S7.2 Prédications sur des programmes</p>	<p>Introduction à la programmation VPL</p> <p>Pair programming</p> <p>Morceler un problème pour pouvoir le résoudre</p> <p>Prédire un comportement, le tester et le modifier</p> <p>Prise en main d'Aseba Studio</p>	
4	<p>S7.1 Programmer le comportement cyan</p> <p>S9.1 Programmer Thymio en mode buzzer</p> <p>S8.2 Le robot fait quelque chose et ils doivent trouver le programme correspondant</p> <p>S9.2 Quiz</p>	<p>Quiz final</p> <p>Morceler un problème pour pouvoir le résoudre</p>	

Tableau 2.2 – Activités sélectionnées pour les séances en classe

Pour les séances données en classe, nous avons décidé d'abandonner complètement l'utilisation du VPL avancé. La raison est simple : il est impossible de compresser douze séances de cours en huit et espérer que les enfants comprennent et retiennent quelque chose. Nous avons donc privilégié la qualité plutôt que la quantité.

Cette nouvelle répartition des activités respecte une division en 4 parties. La première est plus théorique : les enfants y apprennent les notions liées au robot et se familiarisent avec Thymio. La deuxième vise à l'observation et à la compréhension des différents comportements pré-enregistrés de Thymio. La troisième est une introduction à la programmation VPL. Et enfin, la quatrième permet de travailler sur des programmes plus conséquents et d'évaluer les acquis des enfants.

Séance 1

La première séance ne différait pas beaucoup de ce qui avait été prévu puisque c'est à ce moment là que les fondements de la robotique sont expliqués et qu'ils ne peuvent pas être ignorés. Seule l'activité où un enfant prend la place du robot a été abandonnée pour faire place à la formalisation d'un premier comportement. En effet, la formalisation semblait bien plus importante pour la suite des activités (typiquement celles de programmation), alors que les objectifs du jeu de rôle peuvent être acquis lors des observations des différents comportements de Thymio.

Cette séance a rencontré ses objectifs. Elle s'est très bien déroulée dans l'ensemble. Comme nous l'espérions, les enfants étaient super enthousiastes de pouvoir s'approprier le robot et la découverte fut très intuitive. Néanmoins, leur attention a été plus difficile à maintenir lors des moments plus théoriques (activités S1.3 et S2.3). A la fin de la séance, ils ont été capables de fournir des définitions d'un robot très proches de celle établie dans la section 2.1.2 en utilisant les diverses notions apprises au cours de la séance, comme nous le souhaitions.

La discussion du début de la séance a permis de se rendre compte que certains enfants avaient déjà des bases en programmation (avec Scratch essentiellement) et qu'un petit nombre d'entre eux possédaient des robots chez eux. L'un de nos objectifs étant de casser les idées reçues à propos des robots (typiquement celle d'un humanoïde tout en métal qui a des émotions qu'ils nous ont décrit en début de séance), nous avons décidé d'organiser lors de la séance suivante, une petite présentation de leurs robots, pour que les enfants prennent conscience de leur diversité.

La dernière activité de la séance, à savoir, la formalisation d'un comportement, a du être reportée faute de temps. L'une des causes de ce manque de temps est liée à la pédagogie utilisée dans cet établissement. Il s'agit de la pédagogie Freinet¹⁴ qui repose sur le tâtonnement expérimental de l'apprenant. Cette pédagogie se marie très bien avec la séquence de cours puisque nous cherchons à ce que les enfants soient autonomes et curieux. Par exemple, lorsqu'ils rencontrent des problèmes, nous souhaitons qu'ils mettent en place

14. <https://www.icem-pedagogie-freinet.org/node/8309>

des mécanismes de tests et de debugging. Cependant, cette pédagogie offre également une grande liberté aux enfants, jusqu'aux temps de récréation. Il est donc arrivé plusieurs fois que la récréation empiète d'une vingtaine de minutes sur la séance de cours, rendant impossible le fait de suivre le planning.

Séance 2

Pour cette deuxième séance, nous avons prévu de nous focaliser sur l'observation et la formalisation des comportements pré-enregistrés de Thymio. Cependant, comme nous avons décidé à la séance précédente de faire une présentation des robots des enfants, et que nous n'avons pas eu le temps de finir la séance, nous avons dû abandonner d'autres activités. Les activités ayant été abandonnées sont :

- le nommage des comportements (S3.3) car peu utile.
- l'exposition des formalisations des autres comportements (S3.2) car comme ces derniers ne seront pas utilisés, leur utilité est réduite.
- la présentation d'Aseba Studio (S5.2) puisque nous n'avions pas accès à la salle informatique ce jour-là.

Cette séance s'est moins bien passée que la précédente. Les enfants n'étaient pas intéressés par la formalisation des comportements et ils ont eu pas mal de difficulté à comprendre le fonctionnement du pictogramme lié au mouvement et parfois celui lié aux capteurs. Par conséquent, cette activité a duré beaucoup plus longtemps que prévu et nous avons donc dû abandonner l'activité suivante (S4.2) portant sur différents challenges. Celle-ci devait pouvoir être annulée sans conséquence, car l'activité avec les labyrinthes travaillait déjà l'observation des comportements. Quoi qu'il en soit, à l'issue des deux activités de formalisation, la grande majorité des enfants ont compris le fonctionnement des pictogrammes liés aux capteurs et aux actionneurs.

L'activité labyrinthe s'est passée sans encombre. Les enfants étaient hyper enthousiastes et ravis de pouvoir faire part de leurs observations. Et enfin, la discussion à propos de leurs robots, s'est très bien passée également. Les enfants étaient captivés par les différentes démonstrations.

Séance 3

La transition entre activités en classe et activités sur ordinateur nous semblant assez explicite, nous avons délaissé l'activité du déménageur (S5.1) faisant la transition entre les comportements pré-enregistrés et les programmes. Par rapport au planning prévu, la séance 3 a différé, puisque l'on a dû faire la présentation d'Aseba Studio. Cette présentation a pris la place de l'activité de reproduction du comportement rouge (S6.5) puisqu'il avait déjà été formalisé auparavant.

Lors de la présentation d'Aseba Studio et du rappel de la signification des pictogrammes, certains enfants ont montré quelques lacunes de compréhension, celles-ci seront analysées dans la section 2.3.3. Cela aurait été un moment propice à la restructuration avant de les

lancer sur les machines. Cette opportunité manquée retardera un peu la compréhension et l'exécution des exercices. Cependant, après des feedbacks individuels, les enfants ont tous compris et sont tous arrivés au bout de l'activité.

Au fil des activités, de légers soucis se sont présentés. Les enfants éprouvent quelques difficultés pour les exercices théoriques (S6.1 et S6.2) car ils ne perçoivent pas entièrement les différences entre les pictogrammes. L'activité débogging (S6.3) leur pose également problème, car ils n'ont jamais vu de programme plus long, et sont un peu perdus face à un algorithme de plus grande envergure. Par contre, dès qu'ils sont lancés sur les machines, les activités de programmation ne posent dans l'ensemble aucun problème.

Séance 4

Lors de cette dernière séance de cours, nous n'avons pas eu l'occasion de faire l'activité qui consiste à déterminer le programme se cachant derrière un comportement du robot, faute de temps. En effet, alors que l'activité de programmation du comportement cyan (S7.1), qui nous semblait plus complexe, s'est super bien déroulée, l'activité de création du buzzer (S9.1) a pris beaucoup plus de temps que prévu.

Le quiz final fut une réussite avec une majorité de bonnes réponses à chaque question. Les enfants étaient capables de justifier leurs réponses pour expliquer à ceux qui s'étaient trompés. Et nous avons pu également éliminer les doutes subsistant dans l'esprit de certains. De plus, l'idée de se servir du buzzer qu'ils ont programmé eux-même semble les avoir ravies.

Récapitulatif des activités faites

Le tableau 2.3 fait état des activités ayant été réalisées en classe finalement. Dans l'ensemble, même si certaines séances ont été plus fastidieuses, les objectifs de la séquence comme définis dans la section 2.1.1 ont été atteints. Les enfants ont bien perçu le fonctionnement du robot grâce à ses capteurs et actionneurs, et ont compris les concepts de base nécessaires à l'utilisation de la programmation événementielle (les événements, actions, instructions, ...).

Séances	Activités		Compétences et objectifs
	Sigle	Nom	
1	S1.1 S1.2 S1.3 S2.1 S2.3	Qu'est-ce qu'un robot ? Prise en main de Thymio Explications des capteurs, actionneurs, ... Comprendre un des comportements par groupe Rédéfinition de ce qu'est un robot	Introduction au robot Déconstruction des idées reçues sur les robots Définition d'un robot et notions liées Première manipulation de Thymio Appropriation des comportements
2	S3.1a S3.1b S4.1 -	Formalisation d'un comportement tous ensemble Formalisation d'un comportement seul Sortir d'un labyrinthe avec tous les comportements Discussion à propos des robots qu'ils ont chez eux	Formalisation et apprivoisement des comportements de Thymio Formalisation Morceler un problème pour pouvoir le résoudre
3	S5.2 S6.1 S6.2 S6.3 S5.3 S6.4 S7.2	Présentation d'Aseba Studio Enchaînement d'instructions Vrai ou faux Debugging Petites tâches simples à réaliser Le piano coloré Prédictions sur des programmes	Introduction à la programmation VPL Prise en main d'Aseba Studio Utilisation d'autres codes Pair programming Morceler un problème pour pouvoir le résoudre Prédire un comportement, le tester et le modifier
4	S7.1 S9.1 S9.2	Programmer le comportement cyan Programmer Thymio en mode buzzer Quiz	Quiz final Morceler un problème pour pouvoir le résoudre

Tableau 2.3 – Activités finalement réalisées en classe

2.3.2 Améliorations

À l'issue de cette première expérience en classe, deux améliorations majeures sont à prévoir globalement. La première, est de réduire la quantité d'activités que l'on veut faire en une séance. Il vaut mieux prendre son temps pour pouvoir s'assurer de la compréhension des enfants, plutôt que de vouloir tout faire et les perdre en chemin. Autant ne pas atteindre tous ses objectifs, mais veiller à leur maîtrise, au lieu de tous les passer en revue sans qu'aucun ne soit compris. La deuxième amélioration globale consiste à prendre le temps de faire une restructuration à la fin de chaque séance au moins. C'est un moyen de fixer les connaissances acquises durant la séance et cela permet de rectifier la mauvaise compréhension de certains éléments.

Une amélioration possible pour la première séance, serait de donner le robot après avoir donné les consignes. Cela éviterait que les enfants soient dissipés et qu'ils ne retiennent pas tout ce qui leur est demandé, ce qui constituerait un gain de temps.

Pour la deuxième séance, il y a un gros travail à faire au niveau de la formalisation. Commencer par une formalisation en français serait peut-être plus facile. Par exemple, rédiger des phrases du type "Si le capteur devant est allumé, alors je recule" pour le comportement choisi. Et ensuite, diviser ces phrases en deux parties (événement : si [...]; action : alors [...]) et demander aux enfants de les relier. Par la suite, le même type d'exercice pourrait être fait en reliant simplement les pictogrammes préalablement coloriés pour que ce soit plus instinctif.

Pour la troisième séance, l'essentiel des problèmes vient du manque de restructuration et de correction des fautes qu'ils font en répondant aux questions. Cependant, une réorganisation des activités pourrait probablement être bénéfique. Commencer par les exercices de prédictions serait peut-être intéressant puisqu'ils pourraient tester leurs réponses à chaque fois. Enchaîner avec les petites tâches simples à réaliser sur machine, de manière à ce que le fonctionnement des pictogrammes n'ait plus aucun secret pour eux. Poursuivre avec les exercices sur papier (comme le vrai ou faux) et enfin, terminer par le debugging, une fois qu'ils ont une certaine expérience.

Pour la dernière séance, une petite amélioration à faire serait de donner des consignes plus précises concernant la programmation de Thymio en mode buzzer, afin d'éviter que les enfants ne se perdent.

2.3.3 Analyse

La plupart des difficultés rencontrées lors des séances de cours seront très probablement résolues grâce aux améliorations proposées dans la section précédente. Toutefois, l'une de ces difficultés était inattendue et il semble intéressant de s'y attarder un peu plus longuement. Lors de la troisième séance, certains enfants confondaient les comportements pré-enregistrés avec les pictogrammes. Par exemple, en voyant le pictogramme d'événement son, les enfants disaient "C'est le comportement bleu" parce que dans le comportement bleu, Thymio réagit au son.

Sur le moment nous avons pensé que l'enfant faisait juste une analogie, mais par la suite, nous nous sommes rendus compte que le problème était plus important. En effet, les enfants avaient l'impression de programmer des comportements hybrides plutôt que de recréer un comportement en partant de zéro. Cela leur a causé des difficultés dans la suite de la séance. Nous avons remédié à ce problème par quelques feedbacks personnalisés, ce qui a pris pas mal de temps. Finalement, la distinction entre pictogramme et comportement pré-enregistré a bien été intégrée, comme nous avons pu le constater lors du quiz final.

Mais d'où provenait cette confusion ? Sans doute d'un manque d'insistance au cours de la séquence par rapport au programme qui permet de faire fonctionner le robot grâce à une suite d'instructions. Au début, il faut insister sur le fait que robot fonctionne grâce à un programme qui est constitué d'une suite d'instructions. Ensuite, au moment de parler des comportements pré-enregistrés, il faut dire qu'il s'agit de programmes qui ont été écrits par quelqu'un et que l'on peut reconstituer en observant les comportements du robot. Enfin, au moment de leur faire programmer leur propre comportement, il faut réinsister sur le fait qu'on redémarre à zéro. On pourrait éventuellement associer leur premier programme à une couleur non utilisée par les comportements pré-enregistrés pour qu'ils se rendent compte qu'ils peuvent en faire autant qu'ils le souhaitent. En plus d'insister sur ces différents éléments, peut-être faudrait-il réintégrer l'activité du déménageur (S5.1). Cette activité aiderait les enfants à comprendre les limites des comportements pré-enregistrés et par conséquent, l'intérêt d'en programmer de nouveaux. De plus, elle fournirait une bonne occasion d'insister sur les éléments précédemment cités.

2.4 Points d'attention et recommandations pour les enseignants

Pour conclure cette première séquence de cours, nous allons résumer les points importants auxquels il faut être attentif lorsque l'on souhaite effectuer cette séquence.

Tout d'abord, ne pas vouloir en faire trop et penser à restructurer chaque concept vu au cours de la séance. Ceci ne devrait pas être un problème pour les enseignants, car cela fait partie de leurs réflexes.

Au point de vue des concepts, prendre le temps de discuter des idées reçues à propos des robots est vraiment bénéfique pour aider les enfants à s'en faire une représentation générique. Cette représentation les aidera à comprendre la dynamique de la boucle sensori-motrice, car elle est vraiment importante pour comprendre la logique se cachant derrière la programmation événementielle. Par conséquent, insister sur les notions de capteur et d'actionneur est primordial ainsi que les notions d'événement, d'action, d'instruction et de programme.

Enfin, comme expliqué dans la section précédente, il est très important de bien dissocier les comportements pré-enregistrés de Thymio, des programmes que les enfants créeront eux-mêmes.

Le tableau 2.4 présente la séquence adaptée et améliorée que nous referions en classe si nous en avons l'occasion. Elle est toujours constituée de 8 x 50 min de cours et ne couvre donc toujours pas la programmation en VPL avancé. Les séances 1 et 4 restent les mêmes que celles faites en classe. La séance 2 est réorganisée pour permettre aux enfants de se familiariser un peu plus avec les comportements pré-enregistrés du robot, avant de s'attaquer à leur formalisation. Cette activité de formalisation est remaniée pour être plus instinctive pour les enfants. De plus, l'activité du déménageur est introduite en fin de séance pour s'assurer que les enfants fassent bien la transition entre l'utilisation des comportements pré-enregistrés et la programmation d'autres comportements. Enfin, la séance 3 comporte les mêmes activités que réalisées en classe, mais dans un ordre différent pour que les enfants puissent utiliser et tester les pictogrammes avant de devoir les utiliser sur papier.

Thèmes	Sigle	Activités	Timing	Compétences à maîtriser à l'issue de la séance
Introduction au robot	S1.1	Qu'est-ce qu'un robot ?	20 min	Connaître et comprendre les notions suivantes : robot, capteur, actionneur, boucle sensori-motrice, comportement pré-enregistré
	S1.2	Prise en main de Thymio	15 min	
	S1.3	Explications des capteurs	20 min	
	S2.1	Comprendre tous les comportements par groupe	25 min	
	S2.3	Rédéfinition de ce qu'est un robot	20 min	
	S4.1	Sortir du labyrinthe avec tous les comportements	30 min	
Différentiation entre les comportements et les programmes	S3.1	Formalisation des comportements (version améliorée avec du texte ou des pictogrammes à relier)	25 min	Signification des pictogrammes de base Différence entre les comportements pré-enregistrés et les programmes Connaître et comprendre les notions suivantes : événement, action, instruction (Conscience de la diversité des robots existants)
	- OU S4.2	Discussion à propos de leurs propres robots OU Différents challenges	30 min	
	S5.1	Challenge irrésolvable avec les comportements pré-enregistrés	15 min	
	S5.2	Présentation d'Aseba Studio	15 min	
	S7.2	Prédiction sur des programmes	20 min	
Introduction à la programmation VPL	S5.3	Petites tâches simples à réaliser	15 min	Programmation VPL Connaître et comprendre les notions suivantes : instruction et programme
	S6.2	Vrai ou faux	10 min	
	S6.1	Enchaînement d'instructions	10 min	
	S6.4	Piano coloré	15 min	
	S6.3	Débugging	15 min	
	S7.1	Reproduire le comportement cyan	25 min	
Programmation VPL et évaluation	S9.1	Programmer les Thymio en mode buzzer	30 min	Toutes les notions précédentes
	S9.2	Quiz	35 min	

Tableau 2.4 – Séance améliorée selon les observations réalisées

Chapitre 3

Algorithmes et recettes de cuisine

Dans cette séquence de cours, nous avons décidé de faire un parallèle entre des algorithmes et des recettes de cuisine. La première section de ce chapitre explique les objectifs poursuivis par cette séquence, ainsi que les ressources sélectionnées pour y arriver. Ensuite, un plan détaillé des activités est présenté. Contrairement à la séquence précédente, celle-ci n'a pas eu l'occasion d'être testée en classe. Nous expliquons pourquoi dans la dernière section et nous discutons aussi de ce que nous pensons être les points forts d'une telle séquence.

3.1 Objectifs et ressources

Cette section décrit les objectifs poursuivis par cette séquence et en quoi ils s'intègrent aux axes de l'Algorithmique et de la Programmation définis dans la section 1.2. Ensuite, nous expliquons le choix de la programmation par blocs et ce que sont INGINIOUS et Blockly, les deux ressources utilisées pour atteindre nos objectifs.

3.1.1 Objectifs de la séquence

Cette séquence de cours s'inscrit dans les axes Algorithmique et Programmation du référentiel de SI² [9] présentés dans la section 1.2. La partie algorithmique est essentiellement travaillée sur papier. Les enfants sont amenés à lire, comprendre, créer et utiliser des algorithmes simples, leur permettant de résoudre des problèmes divers : des recettes de cuisine, un tri par ordre alphabétique, une calculatrice pour effectuer des règles de trois,... La partie programmation, par contre, se fait sur ordinateur à l'aide d'INGINIOUS et de Blockly qui sont présentés dans la section suivante. Elle s'attarde sur la compréhension du fait qu'un langage de programmation est nécessaire pour être compris par une machine et qu'il faut donc traduire nos algorithmes précédents en langage de programmation, par blocs dans ce cas-ci.

En plus de travailler ces compétences de sciences informatiques, les enfants exercent également des compétences scolaires telles que les arbres de décision, les règles de trois, le travail en équipe, ...

3.1.2 Ressources

Le thème choisi pour cette séquence de cours sur l’algorithmique est la cuisine. Ce choix permet de rendre cette séance accessible à tous, puisque le domaine de l’alimentation concerne tout le monde. De plus, le parallèle entre une recette de cuisine et un algorithme permet aux enfants les plus sceptiques de se rendre compte qu’ils font déjà de l’algorithmique sans le savoir, quand ils préparent un gâteau au chocolat avec leurs parents, par exemple.

Programmation par blocs

Pour les activités portant sur la programmation de la séquence, nous avons décidé de travailler sur base de programmation par blocs. En effet, plusieurs études réalisées par D. Weintrop et ses collègues [17][18][19] ont montré que les élèves étaient plus performants lorsqu’ils utilisaient la programmation par blocs plutôt que la programmation textuelle. Ces études ont été réalisées avec des élèves du début du secondaire. Nous pensons pouvoir appliquer ces conclusions aux élèves de fin de primaire, puisqu’une autre étude de D. Weintrop [20] a montré que les difficultés rencontrées par ces élèves étaient plutôt liées aux concepts abordés qu’à la programmation visuelle.

Cette facilité à travailler avec les blocs est liée à plusieurs facteurs [17] :

- Les blocs sont plus faciles à lire. En effet, ils sont exprimés dans un langage plus verbeux ce qui permet de comprendre rapidement ce à quoi ils servent.
- Le visuel des blocs donne des indices sur les endroits où ils peuvent être utilisés ou non. De plus, les différents emboitements montrent l’importance de l’ordre des instructions. Cependant, il faut veiller à ce que les enfants ne pensent pas qu’il n’existe qu’une seule manière de résoudre un problème avec la programmation, car c’est une idée fautive.
- La composition de programme est plus aisée. La méthode de drag-and-drop permet d’éviter toutes les erreurs de typographie, les oublis d’accolade, de point virgule, et autres. Les élèves perdent moins de temps à trouver où ils ont fait une erreur.
- Les blocs servent d’inspiration. Quand les élèves se retrouvent bloqués sur une partie du code, ils passent en revue les blocs existants pour trouver l’inspiration qui leur permettra de poursuivre.

Les élèves identifient également des problèmes par rapport à l’utilisation de la programmation par blocs, en indiquant qu’ils trouvent cela moins puissant, plus lent (lorsqu’il faut imbriquer énormément de blocs pour faire une condition, par exemple) et parfois moins légitime tout simplement. Ces inconvénients ne perturbent pas cette séquence de cours puisqu’elle vise une simple initiation à la programmation et que les activités reposent sur des programmes relativement basiques.

Blockly et INGIInious

Pour les activités de programmation, nous avons cherché un environnement permettant de tester automatiquement si un programme est correct ou non. De cette manière, les enfants savent directement si leur programme fonctionne ou non et ils ont un petit feedback sur les raisons de leur échec, le cas échéant. Ils sont donc autonomes, ce qui permet à

l'enseignant de s'attarder un peu plus auprès des enfants qui sont en difficulté. Notre choix s'est porté sur INGIInious¹, une plateforme de correction et d'évaluation d'exercices développée au sein du département INGI de l'UCLouvain. INGIInious possède un plugin Blockly², bibliothèque logicielle JavaScript développée par Google, qui permet de créer des environnements de développement utilisant la programmation par blocs. Le résultat de cette intégration est visible à la figure 3.1.

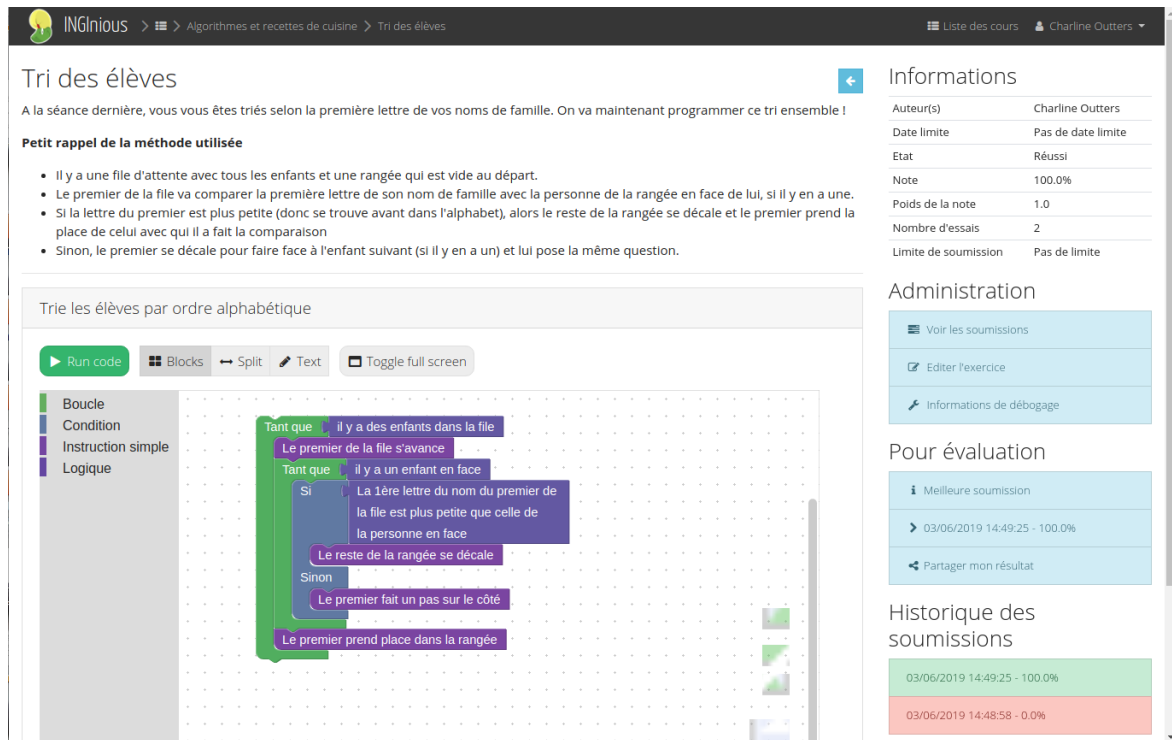


FIGURE 3.1 – Présentation d'un exercice Blockly sur la plateforme INGIInious

Nos tâches INGIInious fonctionnent grâce à trois fichiers :

- `task.yaml` qui contient la configuration de la tâche : son nom, son type, un contexte, le type d'environnement dans lequel elle doit être exécutée, ...
- `run` qui permet d'exécuter le code de l'élève dans l'environnement adéquat et qui retourne si la tâche est un succès ou si elle contient des erreurs.
- un fichier Python, dans lequel le code produit par l'élève est greffé et testé.

Ces trois fichiers sont nécessaires à toutes les tâches sur INGIInious, qu'elles soient faites ou non avec Blockly.

Dans le cadre des tâches Blockly, d'autres fichiers peuvent être ajoutés. En effet, s'il existe déjà un grand nombre de blocs dans la version Blockly de base, il est également possible de créer des blocs sur mesure à l'aide du Blockly Developer Tool³ ou manuellement.

1. <https://inginiuous.org/>
 2. Développé dans le cadre du mémoire de F. Thuin intitulé "INGIInious for kids" : <https://github.com/UCL-INGI/INGIInious-problems-blockly>
 3. <https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>

Cet outil permet de générer un fichier JavaScript contenant d'une part, la description physique du bloc (sa couleur, les entrées du bloc et où elles sont placées, son texte, ...) et, d'autre part, une fonction permettant de générer le code Python pour ce bloc. Une fois ce fichier joint à la tâche, les blocs sont accessibles. Deux exemples de blocs sur mesure sont exposés dans la figure 3.2 ainsi que le code nécessaire à leur création.



FIGURE 3.2 – Exemples de blocs créés sur mesure

Lorsqu'il travaille sur une tâche Blockly, l'élève est face à deux espaces : une toolbox et un workspace (figure 3.3). La toolbox contient tous les blocs pouvant être utilisés par l'élève. Ces blocs peuvent être classés en plusieurs catégories pour que les élèves s'y retrouvent plus facilement. Le workspace est l'espace où ces blocs peuvent être assemblés et seuls ceux-là seront pris en compte lors du test de la tâche. Le workspace peut être prérempli, si nécessaire, pour donner des indices aux élèves ou faire des tâches de réorganisation d'instructions. Les blocs présents dans chacun de ces espaces peuvent être modifiés, soit via l'interface graphique, soit via le fichier `task.yaml` sous forme de XML.

Il est également possible d'ajouter un visuel aux tâches Blockly, pour les exercices avec un labyrinthe dont il faut sortir, par exemple. Grâce à cela, les utilisateurs peuvent tester leur code avant de le soumettre. Pour ce faire, un autre fichier JavaScript sera nécessaire.

Les tâches Blockly réalisées dans le cadre de ce mémoire peuvent être testées dans le cours nommé "Algorithmes et recettes de cuisine"⁴ disponible sur INGINIOUS. Le code de ces tâches est disponible sur GitHub⁵.

4. <https://inginius.org/course/primaire>

5. <https://github.com/lin3out/cuisine-algorithmique>

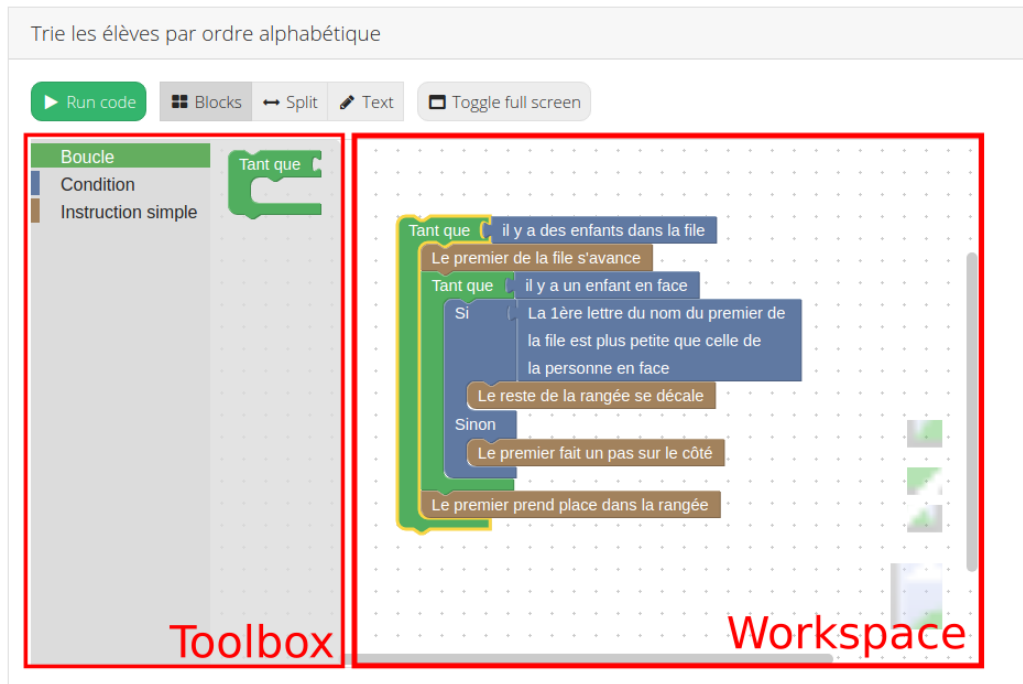


FIGURE 3.3 – Les différents espaces d’une tâche Blockly

3.2 Présentation de la séquence

Pour cette séquence de cours, nous avons préparé cinq séances de 2 x 50 min ciblant les enfants de 9 à 11 ans. Chacune de ces séances s’applique à travailler une des facettes de la programmation à l’aide d’une recette de dessert. Ces facettes sont : la précision des instructions, les instructions induites, les conditions, les boucles et les variables. Une vue d’ensemble de la séquence est présentée dans un tableau récapitulatif, disponible dans la première sous-section, et chaque séance est reprise plus en détail dans la suite de la section.

Ces activités ont été créées suite à ma participation à divers CoderDojos⁶, à diverses lectures (guide 1, 2, 3... *Codez!* [15], archives du concours Bebras⁷, ...) ainsi qu’au test de divers logiciels éducatifs (Algo-Bot⁸, Scratch, ...) qui ont inmanquablement influencé l’élaboration de mes activités.

Tableau récapitulatif de la séquence

Le tableau suivant expose les objectifs, les compétences et les notions étudiées dans les différentes activités ainsi que le matériel nécessaire à chacune.

Les activités colorées en jaune sont les activités déconnectées ; en bleu, les activités qui nécessitent un ordinateur ainsi qu’une connexion à internet ; et enfin, en vert, les activités qui nécessitent du matériel de cuisine.

6. Les CoderDojos sont des ateliers d’initiation à l’informatique pour les jeunes de 7 à 18 ans qui ont lieu une fois par mois. Pour plus d’informations : <https://www.coderdojobelgium.be/fr>

7. Un concours dont le but est de sensibiliser ses participants à l’informatique en tant que science : <https://www.bebbras.be/fr/archives>

8. Un serious game pour apprendre la programmation : <https://www.algobot.be/>

Séances	Activités	Objectifs, compétences et notions	Matériel
Précision des instructions	Analyser la recette Dicter le dessin d'une maison Dresser la table Corriger la recette pour qu'elle soit efficace	Notions d'instruction et de langage de programmation Importance d'être très précis Importance de l'ordre des instructions Application des objectifs précédents	Copies de la recette Dessin basique d'une maison Papier et crayons pour chaque enfant Assiettes, assiettes à entrée, couverts, verres, serviette
	Exécuter la recette	Révision de la recette et amusement	Fromage blanc Sucre Macarons ou rochers à la noix de coco Framboises Ramequins Cuillères
Instructions induites	Algorithme de la tartine Rédiger une recette qu'un robot puisse comprendre Faire les tartines (comme les robots)	Existence des instructions induites Rédaction d'une recette Révision de la recette et amusement	Tartine Pâte à tartiner Beurre Couteau Papier et crayons
	Introduction à Blockly et INGINIOUS Prise en main de Blockly et INGINIOUS Analyse de la recette du glaçage	Familiarisation avec les outils utilisés Notion de condition	Pain Pâte à tartiner Beurre Couteaux Ordinateurs Connexion internet Copies de la recette

Séances	Activités	Objectifs, compétences et notions	Matériel
Les conditions	Arbre de décision du gâteau	Arbres de décision	Copie de l'arbre de décision pour le tableau Copies de gâteaux inventés pour les enfants
	Tri des enfants par ordre alphabétique	Simuler un algorithme en action	-
	Arbre de décision en Blockly	Programmation de conditions imbriquées	Ordinateurs Connexion internet
	Glaçage de cupcake	Révision de la recette et amusement	Cupcakes Citrons Sucre en poudre Bols Cuillères
Les boucles	Analyser la recette des brochettes de fruits	Notion de boucle	Copies de la recette
	Programmer le tri en Blockly	Programmation d'un algorithme de tri	Ordinateurs
	Ecrire le programme pour faire des brochettes	Programmation avec boucles	Connexion internet
	Exécution de la recette	Révision de la recette et amusement	Pommes Ananas Fraises Poires Pics à brochettes
	Analyser la recette du quatre-quarts	Application de ce qui a été vu jusqu'à maintenant	Copies de la recette
	Bref rappel du fonctionnement d'une règle de 3	Règle de trois	Papier et crayons pour les enfants

Séances	Activités	Objectifs, compétences et notions	Matériel
Les variables	Programmation de la règle de trois Calcul des quantités pour la recette	Programmation avec variables Utilisation d'un programme effectué par leurs soins	Ordinateurs Connexion internet Ordinateurs Connexion internet
	Exécution de la recette	Révision de la recette et amusement	Farine Oeufs Sucre Beurre Levure chimique Four Moule à gâteau

Tableau 3.1 – Récapitulatif des activités prévues dans la séquence sur l'algorithmique

3.2.1 Séance 1 - La précision des instructions

Cette séance doit amener les enfants à comprendre ce que sont des instructions pour la machine et à réaliser l'importance de la précision et de l'ordre de celles-ci.

Analyser la recette [15 min]

Les enfants reçoivent la recette inefficace proposée dans l'annexe C. Les instructions y sont peu précises ("Mettre du sucre dans le fromage blanc", la quantité de sucre n'est pas connue) et certaines instructions sont inutiles ("Regarder par la fenêtre"). Sur base de cette recette, nous faisons un parallèle avec l'informatique en déterminant ce qui est équivalent à une instruction. Puis, on leur présente un bout de code basique (en Blockly et en Python) pour leur montrer que les machines fonctionnent avec ces mêmes mécanismes.

Dicter le dessin d'une maison [20 min]

Un enfant est choisi et reçoit un dessin assez basique d'une maison (une porte, deux fenêtres, un étage, un arbre, une cheminée, ...). Le principe est simple : la photocopieuse est tombée en panne et on a besoin de plusieurs copies de ce dessin. L'enfant a donc quelques minutes pour décrire le dessin aux autres pour qu'ils puissent le reproduire. À la fin de l'activité, tous les dessins sont affichés au tableau et les enfants remarquent qu'aucun dessin n'est vraiment pareil. Les enfants réfléchissent à la raison de ces différences et en concluent, avec notre aide, qu'il faut être très précis.

Dresser la table [25 min]

La photo d'une table dressée est montrée aux enfants. Ils ont devant eux le même matériel : une assiette, une assiette à entrée, une serviette, un couteau, une fourchette, une cuillère, une cuillère à dessert et un verre. On leur demande d'écrire l'ordre dans lequel il faut placer les différents objets pour pouvoir reconstituer l'image de départ. Si le timing le permet, ils peuvent échanger leur liste avec leur voisin pour voir si cela fonctionne bien. À la fin, lors de la mise en commun, les enfants remarquent que certaines étapes peuvent être effectuées n'importe quand (placer le verre, par exemple) mais que d'autres ont un ordre impératif (la grande assiette avant l'assiette à entrée).

Corriger la recette pour qu'elle soit efficace [20 min]

Les enfants reprennent la recette inefficace qu'ils ont reçue au début de la séance et la corrigent. Pour cela, il faut retirer toutes les instructions inefficaces (comme "Regarder par la fenêtre") et préciser celles qui doivent l'être ("Dresser le dessert" devient "Remplir la moitié d'un ramequin de fromage blanc", "Saupoudrer de macarons mixés", ...). L'objectif est de s'approcher un maximum de la recette efficace présentée dans l'annexe C.

Exécuter la recette [20 min]

Dernière étape de la séance : réaliser le dessert et le déguster ! Si les enfants se posent des questions sur la manière de faire, cela signifie que leur recette n'est pas assez précise, ils doivent donc la clarifier.

3.2.2 Séance 2 - Les instructions induites

Cette séance de cours traite toujours des instructions et a pour but de montrer que certaines instructions sont induites dans les recettes, mais que cela ne peut pas fonctionner avec un ordinateur. Durant la deuxième partie de la séance, les outils utilisés pour la partie programmation (INGInious et Blockly) sont introduits.

Algorithme de la tartine [15 min]

Nous tenons le rôle de robot et précisons aux enfants que nous ne sommes capables de comprendre que les instructions très simples. Les enfants donnent des instructions pour que le robot puisse se faire une tartine au chocolat. Ils se rendent rapidement compte qu'une instruction telle que "Prendre du chocolat" doit être transformée en "Prendre le bocal de chocolat", "Ouvrir le bocal", "Prendre un couteau" et "Tremper le couteau dans le chocolat". Il y a donc beaucoup d'instructions induites que nous comprenons en tant qu'êtres humains mais que le robot, en tant que machine, ne comprend pas.

Rédiger une recette qu'un robot puisse comprendre [20 min]

Après cette petite démonstration, c'est à leur tour. Les enfants doivent rédiger la recette où toutes les informations sont présentes pour qu'un robot puisse se faire une tartine.

Faire les tartines (comme des robots) [20 min]

Par groupe de deux, l'un joue le robot et l'autre lui fait faire la recette comme il l'a écrite. Quand ils ont fini, ils échangent leurs rôles. S'ils se rendent compte que le robot ne parvient pas à réaliser la recette parce qu'il manque des instructions, ils doivent la retravailler.

Introduction à Blockly et INGInious [20 min]

L'activité commence par une présentation des fonctionnalités de base d'INGInious : créer un compte, comment trouver le cours que l'on cherche, comment trouver la tâche. Ensuite, le fonctionnement de Blockly est expliqué : où se trouvent les blocs, comment ils s'emboitent, ...

Prise en main de Blockly et INGInious [30 min]

Cette séance se termine par un temps pour prendre en main Blockly et INGInious. D'abord, ils passent un moment sur les Blockly Games⁹ pour se familiariser avec Blockly. Le jeu se nommant "Puzzle" est très adéquat comme initiation pour maîtriser la technique d'emboitement des différents blocs. Ils peuvent aussi s'essayer aux autres jeux. Ensuite, les enfants sont invités à se rendre sur INGInious pour effectuer la tâche d'initiation. Le principe n'est pas compliqué, il suffit de réorganiser des instructions, mais cela permet aux enfants de voir comment INGInious réagit en cas de succès ou d'échec. Le résultat attendu de cette tâche d'initiation se trouve dans l'annexe D.

9. <https://blockly-games.appspot.com/>

3.2.3 Séance 3 - Les conditions

Cette troisième séance a pour but d'introduire la notion de condition auprès des enfants. La recette sur laquelle se base cette séance est le glaçage de cupcakes. Cette recette est ultra simple : mélanger du sucre glace à du jus de citron.

Analyse de la recette du glaçage [15 min]

Les enfants reçoivent une recette très courte :

1. Mettre le sucre glace dans un bol
2. Mélanger le sucre et le citron
3. Glacer les cupcakes

Le but de cette activité : comprendre que des conditions sont parfois nécessaires. Par exemple, on ne peut mettre le sucre dans un bol que si ce dernier est propre, ou encore, on ne peut mélanger le sucre et le citron que si le citron a été pressé au préalable.

Arbre de décision du gâteau [15 min]

Pour asseoir la compréhension des conditions par les enfants, nous nous appuyons sur un concept qu'ils ont déjà vu en cours de sciences : la classification grâce à des arbres de décision. L'arbre de décision que nous utilisons est disponible dans l'annexe E et il contient trois niveaux : la couleur du gâteau, la couleur du glaçage et les décorations posées dessus. Les enfants ont devant eux plusieurs images de gâteaux (cf. figure 3.4) et ils doivent déterminer si les gâteaux ont bien été réalisés en suivant l'arbre de décision. Ils sont amenés à justifier leurs réponses, ce qui permet de faire le lien entre une condition et l'arbre.

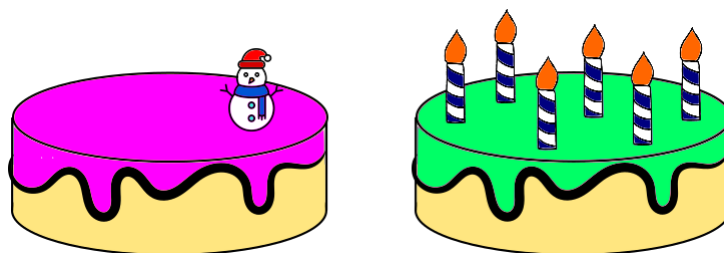


FIGURE 3.4 – Exemple d'images reçues par les enfants (le gâteau correct est en rose et le faux en vert)

Tri des enfants par ordre alphabétique [20 min]

Pour ce tri, nous avons besoin d'une file d'attente (dans laquelle les enfants se placent sans ordre précis) et d'une rangée (où les enfants seront classés par ordre alphabétique).

Les enfants se mettent en file. Le premier se retourne pour commencer la rangée. Le reste de la file a pour consigne de se placer correctement dans la rangée selon la première lettre de leur nom de famille, en suivant un petit scénario :

- Le premier de la file demande au premier de la rangée, la première lettre de son nom de famille.

- Si la lettre du premier de la rangée est située après dans l'alphabet, le premier de la file prend sa place et le reste de la rangée se décale.
- Si la lettre du premier de la rangée est la même ou est située avant dans l'alphabet, le premier de la file recommence le scénario avec le deuxième de la rangée et ainsi de suite.
- Si le premier de la file arrive à la fin de la rangée, il se place à cet endroit.
- Ensuite, c'est au tour du suivant dans la file d'attente.

Au début, les enfants passent un par un, mais une fois qu'ils ont compris, il ne faut plus attendre qu'un enfant soit placé pour envoyer le suivant. Pour terminer, les enfants analysent leur manière de faire. Il y avait chaque fois une comparaison qui permettait d'établir si la condition était vraie ou fausse.

Arbre de décision en Blockly [30 min]

Sur l'ordinateur, les enfants sont invités à refaire l'arbre de décision des gâteaux en Blockly. Comme l'objectif de la séance est la compréhension des conditions, les structures de données sont déjà gérées et ils doivent simplement créer les conditions imbriquées.

Leur programme est testé sur dix gâteaux générés aléatoirement et le message d'erreur leur indique les informations à connaître sur les gâteaux erronés. Cela leur permet de savoir plus facilement où se trouve(nt) leur(s) erreur(s).

Glaçage de cupcakes [20 min]

Comme à chaque fin de séance, il est temps de réaliser la recette corrigée au préalable et de la déguster !

3.2.4 Séance 4 - Les boucles

Cette quatrième séance porte sur la notion de boucle qui est mise à l'honneur grâce aux motifs de brochettes de fruits. Les enfants vont également programmer leur premier algorithme de tri.

Analyser la recette des brochettes de fruits [15 min]

Les enfants reçoivent plusieurs recettes pour faire différentes brochettes de fruits. Dans ces recettes, certaines instructions ou suite d'instructions se répètent ("mettre trois morceaux de fraise puis trois morceaux de pêche" ou "mettre un morceau d'ananas, puis un bout de pomme, puis un morceau d'ananas, puis un bout de pomme,..."). Les enfants identifient les parties se répétant et corrigent la recette en veillant bien à mettre une condition d'arrêt ("répéter 3 fois" ou "répéter jusqu'à la fin du pic à brochette"). Ils ont créé leur première boucle.

Programmer le tri en Blockly [45 min]

Les enfants sont invités à programmer le tri par ordre alphabétique qu'ils ont simulé à la séance précédente. Comme il s'agit du premier programme d'envergure qu'ils réalisent,

l'activité est réalisée tous ensemble. Elle est donc évaluée à 45 minutes pour permettre à chacun de s'exprimer sur les différentes étapes à effectuer. La solution attendue est disponible dans l'annexe F.

Écrire un programme pour faire des brochettes [20 min]

Les enfants écrivent un programme pour créer des brochettes, ils peuvent en faire plusieurs s'ils en ont le temps. Les motifs des brochettes sont libres et les enfants sont vivement encouragés à utiliser les boucles, cependant, la seule condition vérifiée est qu'il y ait bien sept morceaux de fruits par brochette. C'est l'occasion pour eux d'approfondir leur compréhension des boucles. Par exemple, la boucle "tant qu'il n'y a pas 7 éléments sur le pic" ne s'arrêtera pas à 7 éléments s'il y a plusieurs instructions dans la boucle, ils auront donc le message d'erreur indiquant qu'il y a trop de morceaux de fruits sur le pic.

Exécution de la recette [20 min]

Réalisation et dégustation des brochettes de fruits.

3.2.5 Séance 5 - Les variables

Cette dernière séance introduit la notion de variable, en faisant appel à la connaissance des enfants de la règle de trois. Elle nous est utile ici, pour calculer la quantité des ingrédients nécessaires pour faire la recette d'un quatre-quarts.

Analyser la recette pour deux personnes [15 min]

Les enfants reçoivent la recette d'un quatre-quart (cf. annexe G). Après analyse, ils se rendent compte que la recette est complète et précise. Si les enfants ne sont pas d'accord, on peut retravailler la recette jusqu'à ce qu'elle le soit. Par contre, la recette n'est que pour deux personnes, ce qui n'est pas pratique.

Bref rappel du fonctionnement d'une règle de trois [15 min]

Avant de commencer le programme, quelques minutes sont prévues pour faire un rappel du fonctionnement de la règle de trois. Les enfants sont invités à en réaliser quelques-unes sur papier pour se souvenir du mécanisme. Sur base de ce qu'ils ont trouvé, un exemple est affiché au tableau, les enfants identifient les différentes variables nécessaires et réfléchissent au programme qu'ils pourraient écrire.

Programmation de la règle de trois [25 min]

Les enfants vont pouvoir créer une calculatrice pour calculer les quantités d'ingrédients nécessaires. Ils commencent par initialiser des variables pour les ingrédients, le nombre de parts voulues et le nombre de parts prévues par la recette. Puis, ils implémentent le calcul et les quantités nécessaires sont affichées avec le message de succès.

Calcul des quantités pour la recette [15 min]

Grâce au programme qu'ils viennent de créer, les enfants vont pouvoir calculer les quantités nécessaires pour faire le quatre-quart pour tout le monde.

Exécution de la recette [35 min]

Et pour clôturer, les enfants exécutent la recette et dégustent ce délicieux gâteau.

3.3 Discussion

Cette séquence de cours n'a malheureusement pu être testée avec des enfants, car il est difficile de trouver une école acceptant de libérer une dizaine d'heures pour faire ce genre d'activité, dans les derniers mois de l'année. Le rythme étant souvent de 2 x 50 min par semaine, cinq semaines auraient été nécessaires et cela ne rentrerait malheureusement pas dans le planning de ce mémoire. Cette activité sera très probablement testée au sein du CoderDojo de Fernelmont l'année prochaine et peut-être dans d'autres CoderDojo si elle est concluante.

Nous avons bon espoir que cette séquence se déroule sans trop d'accros, car contrairement à la séquence sur la robotique présentée au chapitre précédent, celle-ci ne s'attache à expliquer qu'un seul concept par séance. Lors de la séquence précédente, les difficultés provenaient du manque de temps et du surplus d'informations, mais cela ne devrait pas poser problème ici. Cependant, nous ne sommes pas à l'abri de surprises concernant les incompréhensions des enfants. Par expérience, nous savons que certains enfants ont des difficultés avec l'emboîtement des blocs en programmation visuelle. C'est l'avantage d'utiliser INGINIOUS, car cela va permettre aux enfants les plus doués d'être autonomes et de travailler avec les feedbacks reçus, jusqu'à avoir la bonne réponse. Et pendant ce temps, nous pourrions nous attarder auprès des enfants ayant plus de difficultés pour les aider à atteindre les objectifs.

De manière générale, nous pensons que le mélange entre cuisine et informatique devrait captiver les enfants grâce à la diversité des thèmes et des activités.

Conclusion

A l'heure où le numérique est si présent dans nos vies, il est important que les jeunes soient formés pour pouvoir en tirer le meilleur. Une partie de cette éducation passe par une compréhension de base des sciences informatiques. C'est pourquoi, nous avons développé deux séquences de cours constituant une introduction à certaines facettes de ces dernières.

La première de ces séquences a pour thème la robotique et vise à introduire la programmation événementielle auprès enfants, ainsi que les notions de base utilisées en robotique. La deuxième fait un parallèle entre des algorithmes et des recettes de cuisine, pour qu'ils saisissent les notions de bases de l'algorithmique et qu'ils puissent s'essayer à la programmation.

Le test de la première séquence dans une classe, nous a montré que les difficultés rencontrées par les enfants ne sont pas toujours celles auxquelles on s'attend (la différence entre les comportements du robot et les pictogrammes, par exemple). Malgré ces difficultés, les enfants étaient très contents de prendre part à ce cours et ont acquis les compétences que nous souhaitions leur transmettre. Nous avons tout de même repensé la séquence de cours, afin de faciliter la compréhension de chacun. Toutefois, seule une nouvelle itération de tests pourrait nous indiquer si ces modifications résolvent bien les problèmes rencontrés, sans en introduire de nouveaux.

Malheureusement, le test de ces séquences en classe demande beaucoup de temps, d'organisation et surtout des enseignants acceptant de nous céder quelques unes de leur précieuses heures de cours. C'est pourquoi nous n'avons pu tester, dans le cadre de ce mémoire, qu'une seule séquence, une seule fois. Cependant, nous avons l'intention de réutiliser les séquences créées, entièrement ou partiellement, au sein de CoderDojo et de pouvoir les améliorer par la même occasion.

Dans le futur, il serait intéressant de rendre ces séquences publiques et de les accompagner d'un manuel à destination des enseignants, pour qu'ils soient capables de réaliser les activités dans leur classe sans intervention extérieure. Par exemple, les robots Thymio ont intégrés pas mal d'écoles grâce au projet d'École numérique, et certains enseignants seraient sans doute intéressés d'avoir de nouvelles idées d'activités le concernant.

Bibliographie

- [1] Julie Henry and Noémie Joris. Informatics at secondary schools in the french-speaking region of belgium : myth or reality ? In ***, 10 2016.
- [2] N. Joris and J. Henry. L’enseignement de l’informatique en belgique francophone : état des lieux. In *Actes du colloque Didapro 5 - DidaSTIC (Sciences et technologies de l’information et de la communication en milieu éducatif)*, 2013.
- [3] Stephanie Carretero, Riina Vuorikari, and Yves Punie. DigComp 2.1 : The Digital Competence Framework for Citizens with eight proficiency levels and examples of use. JRC Working Papers JRC106281, Joint Research Centre (Seville site), May 2017.
- [4] The Committee on European Computing Education (CECE). Informatics education in europe : Are we all in the same boat ? Technical report, New York, NY, USA, 2017.
- [5] Modernisering secundair : nieuw model voor studierichtingen – klasse. <https://www.klasse.be/73458/nieuw-model-studieaanbod-secundair/>. (Accès le 29/05/2019).
- [6] F. Heintz, L. Mannila, and T. Färnqvist. A review of models for introducing computational thinking, computer science and computing in k-12 education. In *2016 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, Oct 2016.
- [7] Mark Guzdial. Learner-centered design of computing education : Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6) :1–165, 2015.
- [8] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3) :33–35, March 2006.
- [9] Collectif SI². Proposition de référentiel en sciences informatiques de si². 2017.
- [10] S. Magnenat, F. Riedo, M. Bonani, and F. Mondada. A programming workshop using the robot “thymio ii” : The effect on the understanding by children. In *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 24–29, May 2012.
- [11] F. Mondada, M. Bonani, F. Riedo, M. Briod, L. Pereyre, P. Retornaz, and S. Magnenat. Bringing robotics to formal education : The thymio open-source hardware robot. *IEEE Robotics Automation Magazine*, 24(1) :77–85, March 2017.
- [12] Fanny Riedo, Morgane Chevalier, Stéphane Magnenat, and Francesco Mondada. Thymio ii, a robot that grows wiser with children. pages 187–193, 11 2013.
- [13] Didier Roy, Gordana Gerber, Stéphane Magnenat, Fanny Riedo, Morgane Chevalier, Pierre-Yves Oudeyer, and Francesco Mondada. IniRobot : a pedagogical kit to initiate children to concepts of robotics and computer science. Yverdon-Les-Bains, Switzerland, May 2015.

- [14] Fédération Wallonie-Bruxelles. *Les socles de compétences (toutes les disciplines)*. 2013.
- [15] Claire Calmet, Mathieu Hirtzig, and David Wilgenbus. *1, 2, 3... Codez!*, volume 1 & 2. Le Pommier, 2016.
- [16] Stéphane Magnenat, Morderchai Ben-Ari, Severin Klinger, and Robert W. Sumner. Enhancing robot programming with visual feedback and augmented reality. In *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education, ITiCSE '15*, pages 153–158, New York, NY, USA, 2015. ACM.
- [17] David Weintrop and Uri Wilensky. To block or not to block, that is the question : Students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children, IDC '15*, pages 199–208, New York, NY, USA, 2015. ACM.
- [18] David Weintrop, Heather Killen, and Baker Franke. Blocks or text ? how programming language modality makes a difference in assessing underrepresented populations. 07 2018.
- [19] David Weintrop, Heather Killen, Talal Munzar, and Baker Franke. Block-based comprehension : Exploring and explaining student outcomes from a read-only block-based exam. 02 2019.
- [20] Diana Franklin, Gabriela Skifstad, Reiny Rolock, Isha Mehrotra, Valerie Ding, Alexandria Hansen, David Weintrop, and Danielle Harlow. Using upper-elementary student performance to understand conceptual sequencing in a blocks-based curriculum. pages 231–236, 03 2017.

Annexe A

Formalisation du comportement jaune

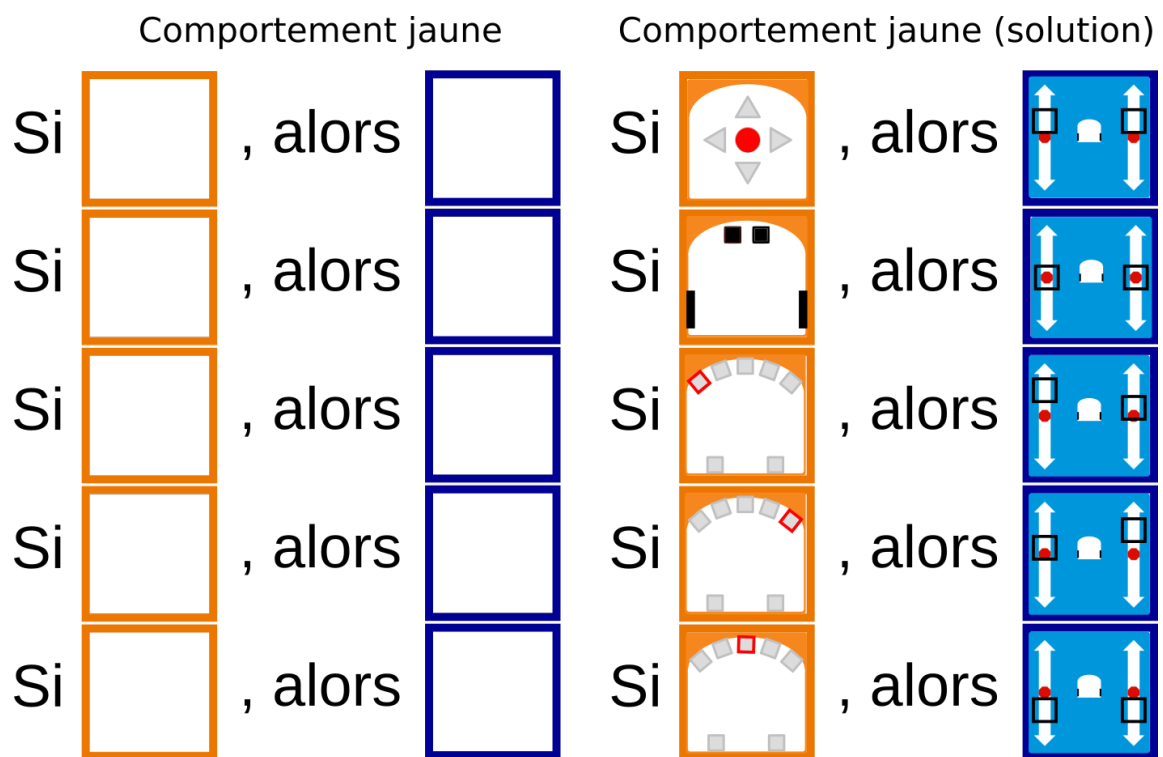


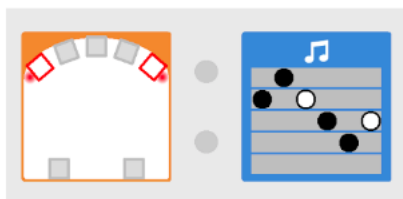
FIGURE A.1 – A gauche, la feuille reçue par les élèves et à droite la formalisation attendue

Annexe B

Exercice de débbugging



Lorsqu'on presse la flèche de devant, Thymio se met en marche et avance. Quand il perçoit des obstacles sur les deux côtés, il joue de la musique. Lorsqu'il voit un obstacle devant, il change de couleur.

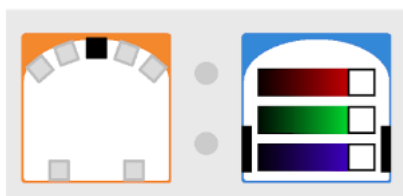


Erreur 1 :

.....

.....

.....

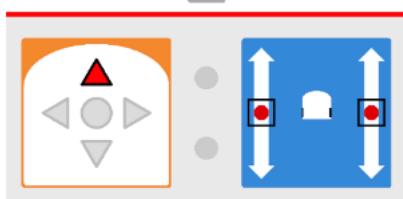


Erreur 2 :

.....

.....

.....



Erreur 3 :

.....

.....

.....

Annexe C

Recette du dessert aux framboises

- Fromage blanc
- Sucre
- Macarons ou rochers à la noix de coco
- Framboises

Recette inefficace

1. Mettre du sucre dans le fromage blanc
2. Boire une verre d'eau
3. Mixer les macarons
4. Regarder par la fenêtre
5. Mixer les macarons
6. Allumer le four
7. Dresser le dessert

Recette efficace

1. Mettre 2 cuillères à café de sucre dans le fromage blanc
2. Mélanger
3. Remplir la moitié d'un ramequin de fromage blanc
4. Mixer une quinzaine de macarons
5. Saupoudrer le fromage blanc de macarons mixés
6. Remplir l'autre moitié du ramequin de fromage blanc
7. Poser un macaron sur le centre du dessert
8. Placer 4 framboises autour du macaron

Annexe D

Tâche d'initiation à INGIInious

Votre réponse a passé les tests ! Votre note est de 100.0%. [Soumission #5cfa43832fff6f5a52fdca8] ×
Bravo ! Tu as réalisé une délicieuse tartine au chocolat !

Remplace les blocs dans le bon ordre

▶ Run code Blocks ↔ Split ✎ Text ◻ Toggle full screen

- Ouvrir le sachet de pain
- Prendre une tartine
- Poser la tartine sur l'assiette
- Ouvrir le pot de chocolat
- Prendre le couteau
- Tremper le couteau dans le chocolat
- Tartiner le chocolat sur la tartine
- Poser le couteau
- Plier la tartine
- Manger la tartine

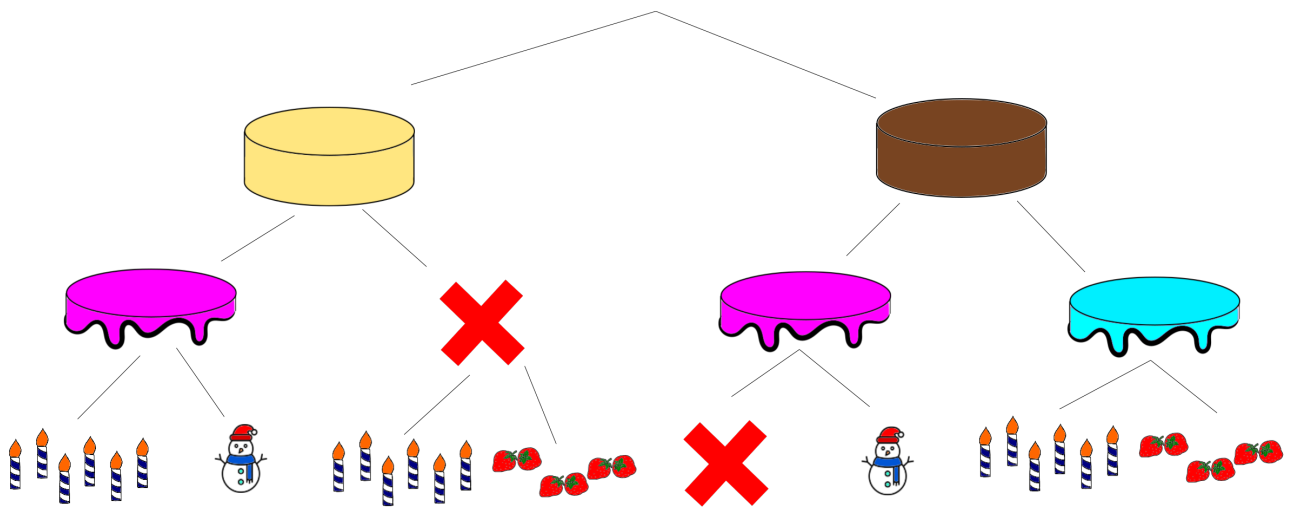
Soumettre >_

Il y a des erreurs dans votre réponse. Votre note est de 0.0%. [Soumission #5cfa43c92fff6f5a52fdcdad] ×
Les instructions ne sont pas dans le bon ordre. Relis bien l'énoncé pour être sûr de l'ordre.

FIGURE D.1 – Au dessus, le résultat attendu avec le message de succès et en dessous, le message d'erreur.

Annexe E

Arbre de décision



Annexe F

Tâche blockly : le tri des élèves

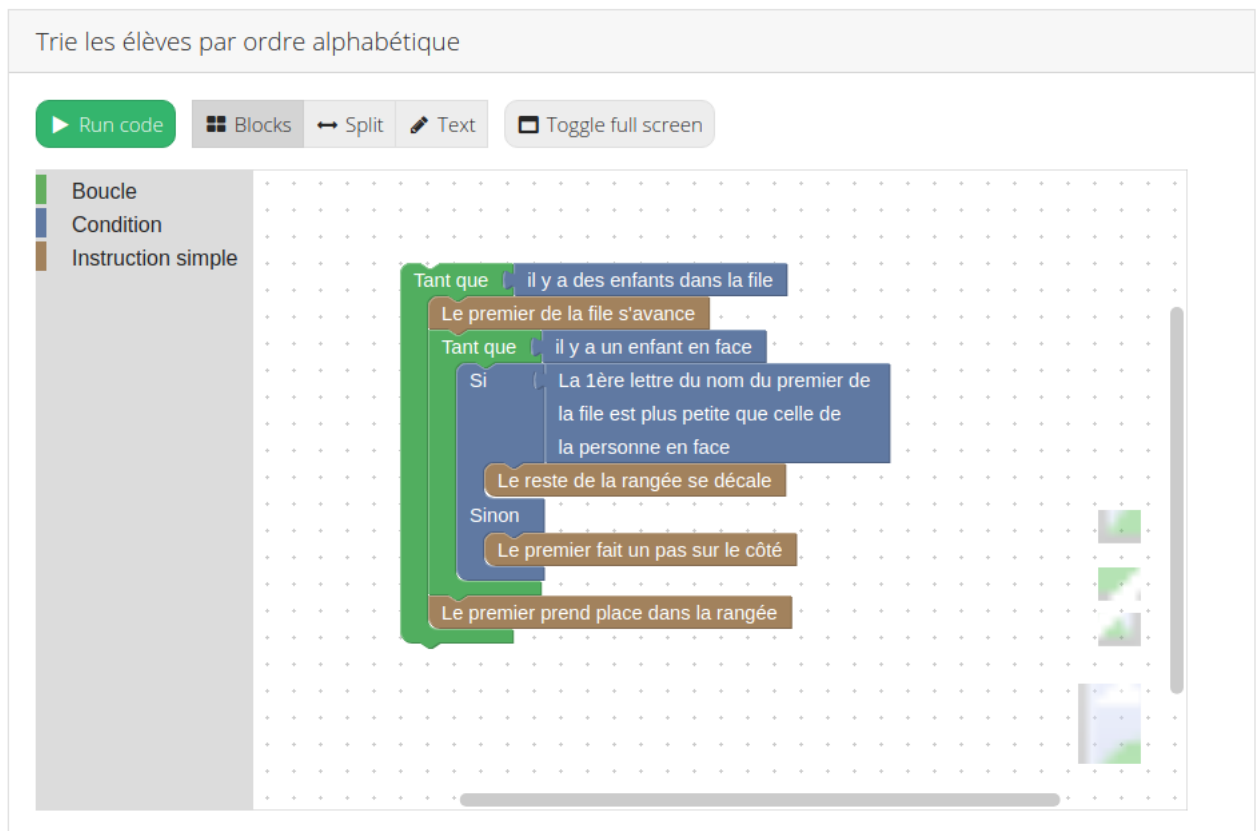


FIGURE F.1 – Résultat attendu pour la tâche de tri des élèves

Annexe G

Recette du quatre-quart

Ingrédients pour 2 personnes

- 1 oeuf
- 45 gr de farine
- 45 gr de sucre en poudre
- 45 gr de beurre
- 3 gr de levure chimique

Recette

1. Préchauffer le four à 180°
2. Mettre l'oeuf dans un bol et le battre au fouet
3. Ajouter le sucre dans l'oeuf
4. Ajouter la farine et la levure chimique tamisées au mélange
5. Mélanger jusqu'à avoir un mélange bien lisse
6. Beurrer et fariner un moule à cake
7. Verser la préparation dans le moule
8. Enfourner pendant 40 minutes

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl