

ALGO-ROBOT : L'APPRENTISSAGE DANS L'ACTION

Julie FRÈVE, professeure en Techniques de l'informatique – Cégep Limoilou

INTRODUCTION

Le projet *Algo-Robot* est né à la suite de l'invitation de la Direction des études du Cégep Limoilou à soumettre des projets liés à des activités d'enseignement et à l'amélioration de la réussite. Dans cette optique, nous avons présenté, grâce à l'appui du Département d'informatique du Cégep Limoilou, un projet visant à mettre en œuvre des notions permettant d'illustrer des concepts liés à la programmation orientée objet et aux algorithmes à partir de la toute dernière génération de robots *Lego*: le *Mindstorms NXT*.

Dans la présentation de son robot, *Lego* mentionnait qu'il était programmable, entre autres, en langage orienté objet *Java*. Le langage *Java* est celui que le Département d'informatique du Cégep Limoilou utilise dans les cours d'initiation à l'algorithmie, dans les cours de développement orienté objet de même que dans certains cours plus spécialisés. C'est ce qui nous a inspirés pour présenter un projet permettant de rejoindre les deux voies de sorties que nous offrons au Département d'informatique de Limoilou, soit l'informatique de gestion et la gestion de réseaux.

1. LE PROJET

1.1 Objectifs

- Explorer les possibilités offertes par l'approche pédagogique constructiviste dans l'apprentissage de l'algorithmie par la mise en œuvre des notions permettant d'illustrer des concepts liés à la programmation orientée objet et aux algorithmes à partir du robot.
- Poursuivre la démarche entreprise par notre département dans la pédagogie des garçons en offrant des situations d'apprentissage dynamiques permettant d'exploiter les capacités d'imagination, de réflexion et d'assimilation des étudiantes et des étudiants.
- Augmenter l'intérêt et la persévérance des étudiantes et étudiants dans leurs études par la nature pratique et tangible des activités offertes avec le robot.
- Offrir des activités où la rétroaction est importante au lieu de procéder par simulations pour ensuite développer la capacité d'abstraction des étudiantes et étudiants.

1.2 Modèle pédagogique

Un philosophe chinois a dit: «Dis-le-moi, je l'oublierai. Montre-le-moi, je m'en rappellerai. Construisons-le ensemble, je comprendrai.»

Les enseignants sont toujours à la recherche de nouveaux moyens pour améliorer les apprentissages de leurs élèves. Depuis quelques années, nous nous questionnions beaucoup sur les méthodes d'apprentissage et sur les mesures et évaluations. Nous observions plusieurs choses, notamment que les évaluations semblaient mieux servir les étudiants moins autonomes. Ceux que nous remarquons pour leur autonomie et pour leurs solutions originales performaient moins bien aux examens. Tout cela nous embêtait beaucoup.

De plus, en participant à la recherche-action auprès des garçons, nous avons approfondi les modes d'apprentissages appliqués aux garçons et constaté que les évaluations étaient peu utilisées dans les cours, du moins au collégial. Ces stratégies n'étaient pas tellement différentes de celles que les filles pouvaient apprécier, sauf qu'il semblait que les filles étaient plus tolérantes à l'immobilité que les garçons.

Nous pensions qu'il fallait offrir plus de diversité dans nos modèles pédagogiques. De plus, nous avons également observé que l'école ne se transforme pas très vite et que la façon d'enseigner des années 2000 est très peu différente de celle des années 60. Le modèle scolaire est toujours bon premier.

Selon la perspective scolaire, l'enseignant organise l'apprentissage dans un ordre logique et la connaissance transmise aux élèves est la même pour tous. Dans ce modèle, le contrôle de l'apprentissage est laissé à l'enseignant. La connaissance est une donnée objective que l'apprenant doit assimiler et repose sur une conception de l'apprentissage qui accorde un rôle déterminant à la relation maître-étudiant.

Le modèle autonomiste origine des sciences cognitives et a pour préoccupation le développement des habiletés et des savoir-faire cognitifs. Ainsi, l'enseignement est conçu de manière à permettre à l'apprenant d'exercer lui-même le contrôle de son apprentissage. Ce modèle accorde une dimension importante à l'apprenant : ce n'est plus l'enseignant qui est le seul détenteur du savoir. L'apprenant doit être actif dans son processus d'apprentissage et en assume lui-même le contrôle.

Nous avons déjà entrepris des remaniements dans nos cours afin d'appliquer de plus en plus le modèle autonomiste et fait certaines expériences qui avaient suscité l'ardeur de nos étudiantes et étudiants. C'est cette ardeur que nous avons voulu provoquer avec le robot. Comme un chanteur ou une chanteuse qui arrive à faire lever les spectateurs et spectatrices, nous recherchions cette poussée d'énergie qui fait que les étudiantes et étudiants s'éveillent, participent et s'engagent dans leur apprentissage.

Juste avant de présenter le projet du robot, nous sommes tombés sur un article sur Pierre Lapointe (auteur-compositeur) qui nous avait habités et fait réfléchir davantage sur nos méthodes d'apprentissage. Voici grossièrement ce que cet article racontait :

L'auteur-compositeur, qui est reconnu pour son grand talent musical raconte qu'il a de la difficulté à concentrer son intérêt sur une seule chose et qu'il est rare que quelqu'un retienne son attention plus de 20 minutes. Jeune, il a toujours pensé qu'il était déficient, car il éprouvait des difficultés d'apprentissage. Au secondaire, par exemple, il était nul dans les cours de musique, alors qu'il composait déjà ses mélodies ! Il était incapable de comprendre la théorie liée à la musique alors qu'il pouvait lui-même réaliser des pièces musicales complexes. Les autres élèves réussissaient le cours haut la main tandis qu'il n'atteignait même pas la note de passage. Ça le rendait triste et il se disait qu'il ne faisait pas partie des personnes intelligentes. Pierre Lapointe est aujourd'hui reconnu comme un créateur polyvalent aux multiples talents et est récipiendaire de nombreux prix tant au Québec qu'à l'étranger.

En présentant le projet du robot, c'est justement ces étudiants que l'on échappe que nous avons voulu accrocher, tous ceux qui ne se sentent pas concernés ou dont l'intelligence est différente et non couverte par nos approches pédagogiques. Nous voulions une alternative à nos méthodes, un lieu d'apprentissage dans la création et l'action où la participation de l'apprenant se révèle dans plusieurs facettes de son apprentissage.

1.3 Nature des activités

- Activités pédagogiques pour les cours de programmation *Java*.
- Activité d'introduction au robot pour les enseignantes et enseignants.
- Formation des enseignantes et enseignants.

1.4 Cours visés

- 420-116-LI—*Algorithmie et programmation orientée objet*
- 420-216-LI—*Développement orienté objet*
- 420-315-LI—*Traitement de données orienté objet*
- 420-473-LI—*Correction de programme*

Au total, ce sont donc quatre cours qui sont directement concernés avec cinq enseignantes ou enseignants par année.

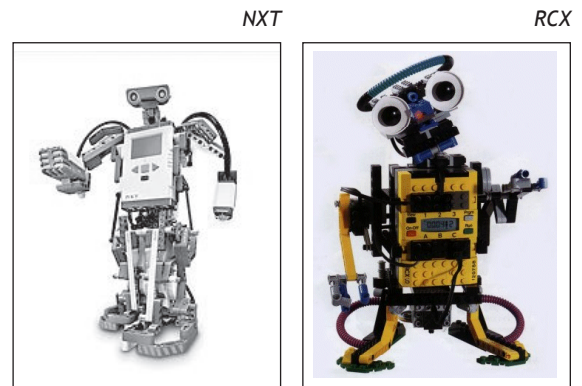
Le robot pourrait également être utilisé dans le cours *Validation d'applications* (420-674-LI) et dans les cours de projets de développement de dernière session où bien des possibilités de développement sont envisageables. On peut penser à la création d'une interface de développement avec *Java* plus conviviale en s'associant avec *SourceForge* ou à un développement d'un jeu interactif avec les robots un peu comme *FIRST LEGO® League. International* le fait mais pour un niveau collégial.

2. PRÉSENTATION DU ROBOT

Dernière génération, successeur du *RCX*.
Janvier 2006

2.1 Principaux composants

- brique programmable *NXT* alimentée par six piles AA et architecturée autour d'un processeur ARM 32 bits, d'un co-processeur AVR 8 bits et d'une capacité de 64 Ko de mémoire;
- trois servomoteurs interactifs, chacun équipé d'un capteur de rotation pour un contrôle précis des mouvements;
- quatre capteurs : lumière, ultrasonique, sonore, de contact;
- haut-parleur;
- afficheur LCD;
- connexion USB et *Bluetooth*;
- sélection de 519 pièces de la série *Lego Technic* pour des constructions robustes et animées;
- logiciel de programmation intuitif dans une version *Lego Labview* et langages compatibles *Java* et *C++*;
- compatible PC et Mac.



2.2 Composants additionnels

Janvier 2007, 3 capteurs additionnels

- boussole;
- capteur de couleurs : permet au robot de distinguer non seulement le noir du blanc, mais aussi une palette de couleurs vives et pastel;
- capteur accéléromètre: permet au robot de se repérer dans l'espace, notamment en reconnaissant le haut du bas et la gauche de la droite.

3. RÉALISATIONS

3.1 Indispensables et contexte

Le projet est né dans la tête de ma fille Charlotte. Il a été intercepté par mon conjoint Pierre Boucher et a cheminé dans la mienne jusqu'à aboutir dans une demande de projet lié à l'innovation pédagogique à la Direction des études de mon cégep.

Lorsque l'aventure du robot *Mindstorms NXT* de *Lego* a débuté, je n'avais pas prévu que nous serions si novateurs. J'ai vite constaté que le robot *NXT* de *Lego* n'intégrait pas le langage *Java* automatiquement. Il fallait développer les bibliothèques et le microprogramme « firmware » nécessaire à l'exécution du langage *Java* sur le robot n'était pas intégré. Je savais qu'une équipe américaine avait construit des applications et les utilisait sur des robots *Lego* de la génération précédente, le *RCX*. L'équipe américaine travaillait à la migration de son travail vers le *NXT*, mais certains problèmes retardaient la distribution de sa réalisation. J'étais dépendante de son travail, car il était impossible pour moi dans le cadre du projet de me lancer dans la construction d'une interface entre le *Java* et l'unité centrale du *NXT*. Le succès du projet reposait sur l'intégration du langage *Java* et du robot, puisque ce langage est celui que nous utilisons à Limoilou pour apprendre aux étudiantes et étudiants les rouages de la programmation orientée objet.

J'ai travaillé en collaboration avec Brian Bagnall, un Américain qui a une passion pour la programmation *Java* et le robot *Lego*. Cet informaticien a démarré un projet libre et développé une version du langage *Java* pour le robot *Lego* qui se nomme « *LeJos* ». Monsieur Bagnall a également écrit la majorité des livres sur les principes de programmation avec le robot *Lego*. Pour toutes ces raisons, il a été mon allié le plus important dans cette aventure. J'ai correspondu plusieurs fois avec lui et grâce à son travail, j'ai pu à partir de la mi-février 2007 programmer en *Java* et exécuter mes programmes à partir du robot *Mindstorms NXT* de *Lego*. Un autre chercheur, anglais cette fois-ci, nommé David J. Barnes, de l'Université de Kent, m'a beaucoup aidée par ses exemples d'application en langage *LeJos*.

Les autres personnes qui m'ont été indispensables sont les suivantes :

- Charlotte Frève-Boucher (10 ans) pour son inspiration et pour la construction des robots;
- Victor Frève-Boucher (7 ans) pour la construction des robots;
- Jean-Simon Aubé (13 ans), Anne-Marie Aubé (11 ans) et Vincent Aubé (8 ans) pour la construction des robots;
- Robert Aubé, professeur au Département d'informatique, pour son appui dans l'achat et la construction des robots;

- Jean-Raymond Gagné, mon mentor, pour toutes ses connaissances si précieuses ;
- Richard Blondeau, technicien en informatique au Département d'informatique du Cégep Limoilou pour son GRAND soutien et sa disponibilité ;
- Pierre-Paul Beaumont et Jean Renaud du service de l'audiovisuel pour leur précieuse collaboration ;
- Les étudiants du cours 420-216-LI à l'hiver 2007 qui ont expérimenté pour la première fois une application *Java* avec le robot; leur enthousiasme est ma plus précieuse récompense ;
- Sophie Lemieux, de la Formation continue et Pierre Boucher, du Département d'informatique, pour leur appui inconditionnel ; ils m'ont permis de ne pas abandonner ;
- Tous mes collègues du Département d'informatique qui m'ont appuyée pour ce projet ;
- La Direction des études sans qui la réalisation de ce projet n'aurait pu avoir lieu et une pensée particulière pour Daniel Boutet qui m'a encouragée à poursuivre.

Voilà! Merci à vous tous et toutes! Vous êtes la preuve qu'ensemble on peut!

3.2 Outils produits

Production d'un guide d'installation et de fonctionnement du robot et du langage.

Les activités d'apprentissage

ACTIVITÉS DU 420-116	
ACTIVITÉ 1	Familiarisation avec le robot, la librairie <i>LeJos</i> et le fichier ANT pour l'exécution des programmes utilisant les librairies <i>LeJos</i> et le robot <i>Mindstorms</i> . Écriture et exécution d'un programme d'essai.
ACTIVITÉ 2	Écriture d'un programme d'une seule classe utilisant un vecteur et permettant de faire bouger le robot.
ACTIVITÉS DU 420-216	
ACTIVITÉ 3	Écriture d'un programme comportant deux classes permettant de faire bouger le robot et le faire réagir à un événement.
ACTIVITÉ 4	Écriture d'un programme permettant d'explorer l'héritage pour faire bouger un robot selon son type et le faire réagir à un événement.
ACTIVITÉS DU 420-315	
ACTIVITÉ 5	Écriture d'un programme permettant à un robot de suivre un trajet précis, semé d'embûches potentielles pour atteindre son but. Le robot pivote sur lui-même s'il rencontre un obstacle. Une deuxième version du même travail est présentée pour le moment où les capteurs de son et visuels seront supportés. Le travail permet d'explorer un objet de la librairie <i>LeJos</i> permettant au robot de se déplacer grâce à ses capteurs de rotation insérés dans ses roues ou de fabriquer sa propre version d'un module de navigation.
ACTIVITÉS DU 420-473	
ACTIVITÉ 6	Correction d'un programme permettant à un robot de saisir une balle et de la déposer à un autre emplacement. La stratégie de déplacement se fait par l'orientation et elle est construite au complet afin de permettre d'inclure des erreurs. <i>LeJos</i> offre la classe <i>TimingNavigator</i> qui fait essentiellement le même travail.

3.3 Étapes de réalisation

MONTAGE D'UN PROTOTYPE DU ROBOT

Étude des modèles les plus adaptés pour le travail à réaliser avec nos étudiantes et étudiants et les plus faciles à manipuler.

Le premier modèle expérimenté, l'humanoïde, éprouve des problèmes de stabilité : il tombe facilement.

NXT



Le modèle araignée est intéressant, mais les pinces qui servent de bras sont accrochantes et faciles à désarticuler. On peut prévoir ajouter les pinces dans des cours plus avancés.

Spike



Le modèle *Tribot* intègre tous les capteurs mais il offre les avantages d'être peu encombrant, peu problématique et plus stable (difficile à démolir ou facile à reconstruire).

Tribot



Dans nos choix, il fallait tenir compte des possibilités du robot, de son attrait pour les étudiantes et les étudiants ainsi que de la facilité au niveau logistique pour les enseignantes et enseignants. L'humanoïde était sans doute le modèle offrant le plus de possibilités de programmation, d'attraits pour les étudiantes et étudiants, mais d'une difficulté extrême d'un point de vue logistique. Il fallait déconstruire une partie du robot pour changer les batteries, la prise USB était difficile à atteindre, les bras se décrochaient facilement lors du transport, etc. Le modèle *Tribot* a donc été retenu, dans une version allégée pour les premiers cours (sans les bras), mais ceux-ci peuvent être ajoutés par la suite.

TESTS DE PROGRAMMATION

Ici commencent les difficultés!

- *iCommand* et *Bluetooth*
- *LeJos* non adapté pour *NXT*

Montage des six robots

Recherche d'une solution *Bluetooth* et abandon

Version *LeJos NXT*

- Résolution des problèmes d'installation et d'exécution
- Résolution du trombone
- Résolution de l'intégration (ANT)
- Résolution de problèmes de robotique

EXPÉRIMENTATION EN CLASSE

L'activité d'apprentissage 2 a été expérimentée auprès de deux groupes de 420-216. Cette activité s'est déroulée sur deux séances de laboratoire de trois heures. L'un des groupes était sous ma

responsabilité et j'ai accompagné l'enseignant de l'autre groupe. L'expérimentation a été le point culminant. L'intérêt était évident et les activités ont été réalisées avec beaucoup d'enthousiasme. Quelques problèmes sont cependant survenus au niveau du temps de recharge des batteries et des difficultés dues à une difficulté de maîtrise de l'environnement de développement *JDeveloper* par les étudiants. À peu près tous les défauts de fonctionnement du robot étaient dus à la création d'un projet *JDeveloper*.



Classe

FORMATION

Formation des enseignantes et enseignants (1 journée) où les activités d'apprentissage ont été présentées, l'installation effectuée et les tests effectués. Les enseignantes et enseignants sont maintenant en phase d'appropriation en exécutant les activités d'apprentissage.

ET L'AVENIR ?

Possibilités d'utiliser le robot dans les cours de projet, de participer à des concours de développement à partir du robot, de développer une interface de développement plus conviviale, d'ajouter des composants au robot, etc.

Le groupe de développement de M. Bagnall continue d'améliorer son produit *LeJos*. Il est primordial de garder le contact afin de profiter de ses améliorations. Au moment de terminer ce rapport, le livre de M. Bagnall sur la dernière version *LeJos* et du *NXT* est disponible.

Meilleure intégration, plus de composants supportés (son, ultrasonique, couleur, *Bluetooth*, etc).

Notons que l'Université de Genève, l'Université du Kent en Angleterre et La Sierra University en Californie sont quelques-unes des institutions à utiliser le robot *Lego* pour l'apprentissage de la programmation ou pour réaliser des Olympiades dans leur domaine d'études.

En 1980, le mathématicien Seymour Pappert du Massachusetts Institute of Technology (MIT) et inventeur du langage *LOGO* a établi sa « géométrie de tortue » pour les enfants en substituant la notion d'une tortue à celle du point; la tortue étant une entité qui a une position et une direction. Ce même mathématicien a identifié la puissance évocatrice du modèle de robot *Lego* et en a fait part à la communauté dans un livre. Plusieurs expériences plus tard, la nôtre s'inscrit dans ce courant en s'appuyant sur les résultats de notre recherche-action sur la pédagogie des garçons.

CONCLUSION

Foi / Ardeur / Persévérance / Soutien

MÉDIAGRAPHIE

BAGNALL, B., *Core Lego Mindstorms Programming*, Upper Saddle River, Prentice Hall, 2002. TJ211.B147 2002.

BAGNALL, B., *Maximum LEGO NXT*, mai 2007.

BAGNALL, B., *Programmation des robots Lego Mindstorms*, Paris, Campus Press, 2002. TJ211.B1471 2002.

BARNES, D. J. *An API for the Lejos platform, intended for introductory Java instruction. Online. Internet.* [En ligne] WWW: <http://www.cs.ukc.ac.uk/people/staff/djb/rcx/>. (August 22, 2001).

BARNES, D. J., *Teaching Introductory Java through LEGO MINDSTORMS Models*, Canterbury University, 2002.

BAUM, D., *Definitive Guide to LEGO MINDSTORMS*, Apress, 2000.

Lego, site officiel. [En ligne] <http://mindstorms.lego.com>.

LeJos Java Mindstorms, site officiel pour les mises à jour et les nouveautés livresques. [En ligne] <http://lejos.sourceforge.net>.

KNUDSEN, J., *The Unofficial Guide to LEGO MINDSTORMS Robots*, O'Reilly, 2000.

An API for the leJOS platform intended for introductory Java instruction. [En ligne] <http://www.cs.kent.ac.uk/people/staff/djb/rcx/>.

First Lego League. [En ligne] <http://www.legoleague.org/fr/>.

Wikipédia. [En ligne] <http://www.nxt-mindstorms.com/wiki/Category:NXT>.