

CodeLab IDE & Simulators

Un outil pédagogique pour apprendre à coder avec bien plus que des interactions écran-clavier

Jérôme Lehuen

LIUM, Le Mans Université, France
lium.univ-lemans.fr

Abstract: L’atelier que nous proposons vise à mettre en œuvre et à utiliser CodeLab, un IDE ouvert à base de *plugins applicatifs* (modules) dédié à l’apprentissage de la programmation. Un module CodeLab est l’association d’une IHM et de plusieurs APIs disponibles pour chacun des langages cibles (C, Python, Java, *etc.*). Les apprenants peuvent travailler les concepts algorithmiques et le codage tout en bénéficiant d’applications cibles possédant des caractéristiques riches et motivantes (graphisme, robotique, simulations, *etc.*). CodeLab est fourni avec un kit de développement permettant à tout développeur Java de concevoir et de produire des nouveaux modules applicatifs. Par ailleurs, CodeLab est utilisable de façon autonome ou bien connectée : un serveur dédié (également diffusé) qui héberge des sessions (classes virtuelles) permet à un tuteur de suivre en temps réel le travail des apprenants, de compiler et/ou tester leurs programmes, de communiquer avec eux par l’intermédiaire d’une messagerie instantanée, voire de prendre le contrôle de leur espace de travail, afin d’intervenir sur leurs programmes.

Keywords: IEIAH, IDE extensible, plugin, codage, classe virtuelle.

1 Description de CodeLab

1.1 Un IDE ouvert à base de *plugins applicatifs*

CodeLab [1] est un environnement pédagogique dédié à l’apprentissage de la programmation multi-langages (Python, Java, Langage C, *etc.*) et multiplateformes. Son originalité est de proposer, en plus des fonctionnalités d’un IDE classique, des alternatives ludiques ou techniques aux traditionnelles interactions écran-clavier pour la conception des activités pédagogiques, au travers de l’utilisation de modules applicatifs. Ces modules, disponibles sous la forme de *plugins*, peuvent être des visualisations, des panneaux de contrôle, des simulateurs, et autres systèmes temps-réel pilotables par programme. Chaque module est l’association d’une IHM et de plusieurs APIs (*Application Programming Interfaces*) disponibles pour chacun des langages cibles. Les apprenants peuvent ainsi travailler les aspects algorithmiques et le codage, tout en bénéficiant d’applications cibles possédant des caractéristiques riches et motivantes (cf. Fig. 1).

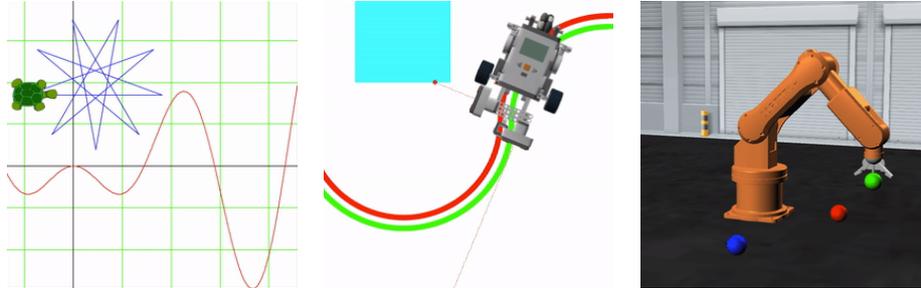


Fig. 1. Exemples de modules applicatifs CodeLab

D'un point de vue pédagogique, notre approche s'inspire du courant constructiviste et des théories sur l'apprentissage de Jean Piaget [2] en France, de Marvin Minsky et de Seymour Papert [3] [4] aux USA. Ce dernier a donné naissance au langage Logo et à sa célèbre tortue, ainsi qu'à la gamme robotique Mindstorms issu de sa collaboration avec le groupe LEGO™. C'est la raison pour laquelle le premier module de CodeLab comporte une tortue « *Logo-like* » et le second un simulateur de robot LEGO NXT. CodeLab généralise le principe en proposant un kit de développement de modules qui permet à tout programmeur Java de concevoir et de développer des nouveaux modules applicatifs. L'utilisateur final n'a qu'à glisser-déposer le *plugin* (sous la forme d'un fichier .pac) dans le dossier « modules » pour bénéficier immédiatement d'un nouveau module applicatif (IHM + APIs) accessible dans un nouvel onglet. Les premiers modules CodeLab développés et distribués à ce jour sont les suivants :

- Traceur graphique 2D à base de primitives + tortue graphique ;
- Simulateur de robot mobile LEGO NXT (capteurs configurables) ;
- Simulateur de robot industriel 5 axes en 3D (avec préhension et tracé) ;
- Simulateur de moteur NXT équipé d'un capteur de rotation ;
- Visualiseur (diagramme à barres) pour algorithmes de tris.

Le kit de développement de modules comporte un ensemble de classes Java à hériter, ainsi que des scripts de génération. Par exemple, les bibliothèques à inclure (C, Python, *etc.*) sont générées automatiquement à l'aide d'un script, à partir d'une description XML de l'API du module (constantes, instructions, fonctions, méthodes).

1.2 Mise en œuvre des classes virtuelles

Une autre fonctionnalité de CodeLab, reprise du logiciel Hop3x [5] [6] mais également présente dans quelques IDEs en ligne comme Replit [7], est la constitution de classes virtuelles (en présence ou à distance) grâce à une architecture client-serveur dédiée. Le modèle utilisateur / groupe / session permet la constitution de groupes de TP à géométrie variable, encadrés par un ou plusieurs tuteurs. Ces derniers peuvent suivre en temps réel le travail des apprenants, compiler et/ou tester leurs programmes, communiquer avec eux par l'intermédiaire d'une messagerie instantanée intégrée, les déconnecter en

fin de séance, *etc.* Par ailleurs, les apprenants peuvent « lever la main » (bouton) et attendre qu'un tuteur engage une discussion *via* la messagerie instantanée.

L'interface « tuteur » de CodeLab (cf. Fig. 2) est une spécialisation de l'interface élève présentant la liste des apprenants (connectés ou non) d'une session donnée, qui permet d'accéder à leur espace de travail, ainsi qu'à diverses informations les concernant (heure de connexion, adresse IP, demande d'aide, *etc.*). Elle permet également de prendre la main sur l'espace de travail d'un apprenant, lorsque ce dernier est en ligne.

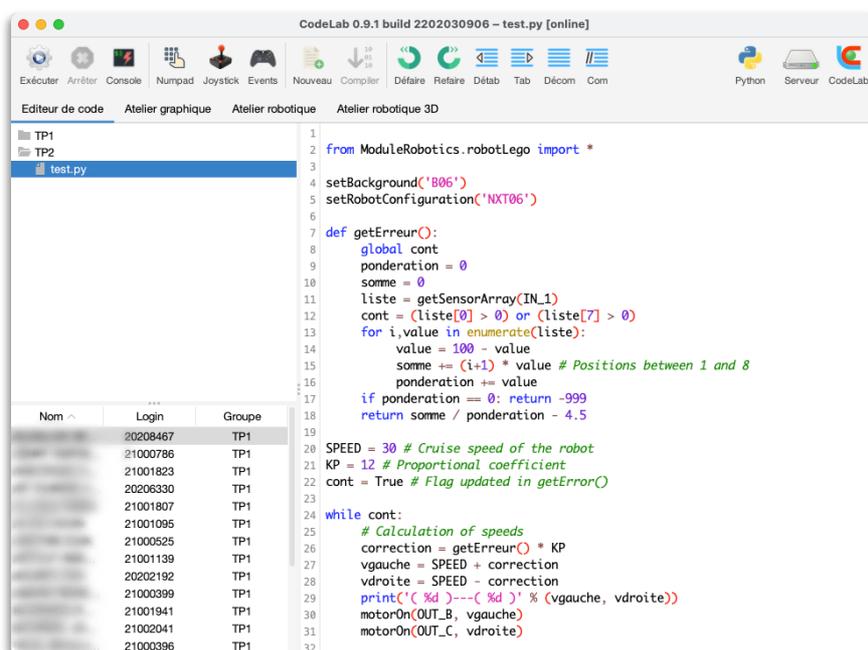


Fig. 2. Interface de CodeLab en mode tutorat de classe virtuelle

1.3 Un langage à base de blocs pour introduire Python

CodeLab a vocation à être utilisé avec des langages professionnels comme C, Python, Java, *etc.* mais, afin de cibler un public plus jeune, nous avons également intégré un langage par assemblage de blocs qui permet de découvrir les structures de programmation en s'abstrayant d'une syntaxe formelle. Programmer consiste alors à déplacer et à assembler des blocs à l'aide de la souris, et à compléter les instructions grâce à des fenêtres de dialogue. Une fois la structure de blocs construite, il est possible de l'exécuter, mais aussi d'en visualiser une traduction en langage Python, à l'intérieur-même des blocs assemblés (cf. Fig. 3). Cette fonctionnalité, également présente dans l'environnement Scratch [8], a pour objectif de démystifier les langages fondés sur des grammaires formelles en permettant une transition douce depuis un langage graphique plus ludique, vers un langage grammatical comme Python.

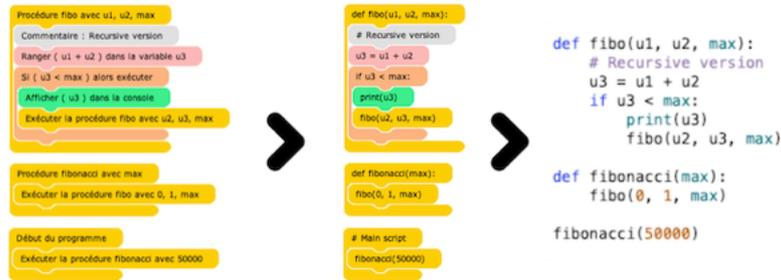


Fig. 3. Transition du langage « Blocs » vers le langage Python

2 Activités proposées dans le cadre de l'atelier

Dans le cadre de cet atelier, nous proposons de réaliser avec CodeLab quelques activités basées sur les modules « Atelier Graphique » et « Atelier Robotique » en Blocs, Python ou langage C. Le groupe sera associé à une session de classe virtuelle, ce qui permettra également d'illustrer les possibilités de tutorat. Nous pourrions adapter ou enrichir les activités suivantes, en fonction des demandes des participants :

- Activités de programmation dans le module de tracé graphique 2D ;
- Activités de programmation du robot NXT du module robotique ;
- Mise en œuvre et tutorat d'une séance de TP en classe virtuelle ;
- Installation d'un serveur CodeLab sur une machine perso (éventuellement).

Les participants pourront également installer CodeLab et un serveur CodeLab sur leurs machines personnelles s'ils le désirent.

Références

1. Site web du logiciel CodeLab : <https://codelab.univ-lemans.fr>
2. Piaget, J.: La Naissance de l'intelligence chez l'enfant. Neuchâtel, Delachaux et Niestlé (1936, 1970).
3. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, Inc., New York (1980).
4. Papert, S.: The Conservation of Piaget: The Computer as Grist. Constructivism in the computer age, George Forman, Peter B. Pufall (eds.) pp. 3–14 (1988).
5. Site web du logiciel Hop3x : <https://lium.univ-lemans.fr/hop3x/>
6. Lekira, A., Després, C., Jacoboni, P. *et al.* Using Indicators during Synchronous Tutoring of Practical Work. The 11th IEEE International Conference on Advanced Learning Technologies (ICALT'11), Athens, Georgia, USA. pp.568–572 (2011).
7. Site web du logiciel Replit : <https://replit.com>
8. Site web du logiciel Scratch : <https://scratch.mit.edu>