

Proglab : l'algorithmique accessible

François Pirsch(*)

Quel langage choisir pour pratiquer l'algorithmique au lycée ? Celui des calculatrices ? Pascal ? Python ? R ? Maxima ? Scilab ? Il y en a des dizaines⁽¹⁾, on a l'embarras du choix. Il y a une telle variété de syntaxes ou de domaines d'application qu'il y en a pour tous les goûts. En même temps on se heurte très vite à des difficultés : dans certains cas l'installation du logiciel est problématique, dans d'autres c'est la syntaxe pleine de symboles et de mots-clés bizarres qui bloque les débutants dès le départ. Parfois aussi c'est le processus écriture – compilation – exécution qui est trop long.

Le logiciel Algobox⁽²⁾, de Pascal Brachet, a résolu une bonne partie de ces difficultés : une installation sans histoires (il y a même une version portable sur clé USB, si on n'a pas la possibilité de l'installer dans l'établissement) une syntaxe en français, une interface bien adaptée à de grands débutants, des fonctionnalités explicitement dédiées aux nouveaux programmes du lycée. Ces qualités l'ont fait rapidement et largement adopter dans la communauté éducative.

Mais l'installation d'un logiciel ad hoc sur les machines du lycée, et sur celles des élèves si on veut qu'il puissent réaliser des travaux personnels, reste toujours un obstacle. En créant le site <http://proglab.fr>, j'ai voulu permettre une utilisation directement en ligne, sans rien installer. Avec en plus la possibilité de travailler dans plusieurs langages.

Simplifier l'accès et augmenter le confort d'utilisation

Cette première idée directrice est purement pragmatique : avec les horaires actuels, et compte tenu du poids de l'algorithmique dans les programmes, une certaine fluidité est nécessaire dans le maniement concret des langages. Il est important d'économiser des manipulations, des questions techniques pas forcément directement en rapport avec l'exercice du jour, pour pouvoir se concentrer sur l'essentiel. Proglab essaie d'aller dans ce sens : ce n'est pas un logiciel mais un service en ligne. Il n'y a rien à installer, on va sur le site et on peut travailler, sans inscription ni mot de passe, directement. Les élèves peuvent ainsi travailler sur n'importe quel ordinateur, pourvu qu'ils disposent d'une connexion internet.

Bien sûr, l'utilisation en ligne soulève le problème d'une éventuelle coupure de connexion. La plupart des sites qui permettent de programmer en ligne⁽³⁾ reposent sur des échanges permanents avec le serveur : en cas de défaut de connexion, ou de

(*) fpirsch@proglab.fr

(1) Wikipedia en compte 662

(http://fr.wikipedia.org/wiki/Liste_des_langages_de_programmation)

(2) <http://www.xmlmath.net/algobox/>

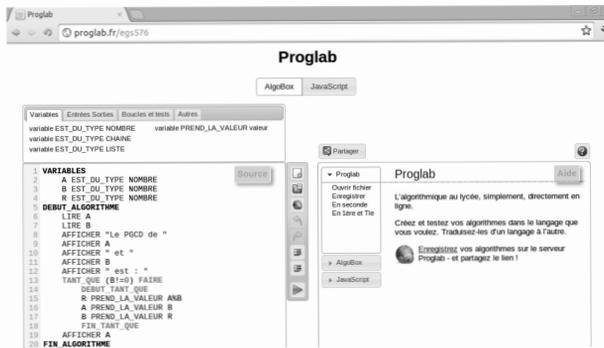
(3) À noter, le logiciel de calcul formel Xcas en ligne :

http://www.xcasenligne.fr/giac_online/demoGiacPhp.php

maintenance même de quelques minutes, il devient impossible de travailler. Sur Proglab, passé le chargement initial de la page, les fonctionnalités importantes (écrire, lancer, traduire l'algorithme) ne nécessitent plus de passer par le serveur, et on peut continuer à travailler hors ligne.

L'interface est assez dépouillée, mais son apprentissage est très rapide et malgré la simplicité d'accès, l'éditeur a quand-même toutes les fonctionnalités indispensables :

- La structure du programme est mise en évidence par des couleurs qui distinguent les éléments : tests en violet, boucles en orange, instructions en bleu, ...
- Des suggestions pendant la frappe permettent de compléter automatiquement les mots commencés, et ainsi d'éviter les erreurs de frappe tout en gagnant du temps.
- Les erreurs sont signalées au fur et à mesure de la frappe et sont accompagnées de suggestions de correction.
- On peut augmenter ou diminuer le retrait d'un bloc entier d'instructions d'un simple clic.
- Il est possible à tout moment d'annuler et de refaire toutes les actions.
- Ces fonctionnalités sont accessibles à la fois en cliquant sur des icônes et par des combinaisons de touches au clavier : la main n'est pas obligée de passer sans arrêt du clavier à la souris.



L'écran est divisé en deux parties. L'algorithme une fois écrit dans la partie gauche, son exécution est immédiate d'un simple clic et la sortie (textuelle, graphique ou même animée), apparaît dans la partie droite. On peut ainsi tester en cours d'écriture, suivre des modifications, compléter ou corriger des affichages.

Utiliser plusieurs langages

La deuxième idée directrice est plus philosophique, c'est de permettre de travailler dans plusieurs langages. Il serait dommage de s'enfermer dans un seul langage de programmation. À l'issue de la conférence devLink 2012, Jim Cowart⁽⁴⁾

(4) Développeur de haut vol, participant à plusieurs grands projets open-source, et blogueur : <http://freshbrewedcode.com/jimcowart/2012/09/01/is-JavaScript-tragically-important-2/>

résume sa vision du développement informatique par cet algorithme :

1. Efforcez-vous de faire ce que vous aimez.
2. Visez l'excellence dans ce que vous faites.
3. Ne limitez pas vos compétences à ce que votre employeur utilise actuellement.
4. Apprenez un autre langage.
5. Répétez l'étape 4.

Pour l'instant Proglab propose deux langages :

- Algobox, pour ses qualités et sa notoriété au lycée
- JavaScript parce qu'il est partout⁽⁵⁾, en particulier dans tous les navigateurs web.

Le passage d'un langage à l'autre se fait à volonté, instantanément en cliquant sur le langage voulu. C'est Proglab qui se charge de la traduction du programme en cours. Cette opération est loin d'être triviale, et il est intéressant de constater les différences de syntaxe⁽⁶⁾.

En effet même pour des programmes simples, la traduction ne peut pas être littérale et demande de résoudre divers problèmes qui ne manquent pas d'intérêt et permettent de faire réfléchir les élèves sur la question des langages.

Par exemple dans Algobox, pour entrer une chaîne de caractères, on déclare d'abord la variable de type chaîne :

```
s EST_DU_TYPE CHAINE
```

Puis, plus loin dans le programme, on demande son entrée.

```
LIRE s
```

Le logiciel Algobox affiche alors le message « Entrez s » et demande la valeur.

En JavaScript on ne déclare pas le type des variables. D'ailleurs les types de variables sont dynamiques, ce qui est très pratique dans certains cas (bien au-delà du programme du lycée).

On utilise la fonction `prompt` qui demande à l'utilisateur d'entrer du texte, mais qui nécessite de donner explicitement le message d'invite. Dans notre cas cela donnerait

```
s = prompt("Entrez s");
```

Voilà un cas où une information implicite dans un langage doit être rendue explicite au moment de la traduction dans un autre langage. Pour la traduction inverse, il faut supprimer cette information.

Si c'est d'un nombre qu'on a besoin, les choses se compliquent un peu. Dans Algobox, on déclare la variable de type nombre :

```
n EST_DU_TYPE NOMBRE
```

(5) http://www.lemonde.fr/technologies/article/2012/02/08/javascript-le-vilain-petit-canard-devenu-couteau-suisse-du-web_1640122_651865.html

(6) Les différences d'esprit ne se voient pas vraiment quand on reste dans les limites du programme de lycée. On peut se faire une idée des syntaxes en regardant le même programme trivial qui affiche le message « Hello world ! » écrit dans des dizaines de langages différents : http://fr.wikipedia.org/wiki/Liste_de_programme>Hello_world

Puis, plus loin dans le programme,

```
LIRE n
```

Si on applique la recette précédente, cela donne en JavaScript :

```
n = prompt("Entrez n");
```

Mais dans ce cas il y a un bug : le résultat de la fonction `prompt` est une chaîne de caractères, et si l'utilisateur tape le nombre 54, la variable `n` aura pour valeur la chaîne de caractère « 54 ». Cela peut donner des résultats surprenants, par exemple `n+1` vaudra « 541 » car le `+` sera compris comme la concaténation de deux chaînes de caractères. Il faut donc expliciter la conversion en nombre en utilisant la fonction `parseFloat` (en anglais, « analyser comme un nombre à virgule flottante⁽⁷⁾ ») :

```
n = parseFloat(prompt("Entrez n"));
```

On a donc là un cas où une même instruction d'un langage (LIRE) a différentes traductions selon les circonstances. Dans d'autres cas, il faut introduire des variables supplémentaires. Parfois il est impossible de traduire sans réécrire entièrement tout le programme. Pour les algorithmes du lycée, heureusement, la traduction est relativement simple. Proglab tient compte de ces petites subtilités de manière à ce qu'à n'importe quel moment on puisse faire des allers-retours entre les langages tout en gardant un programme valide et exécutable. Pour cela il doit parfois introduire des modifications mineures.

Je prévois de faire évoluer le site pour que d'autres langages puissent être utilisés par la suite, en particulier Python⁽⁸⁾. Largement utilisé, bien adapté à un usage pédagogique et scientifique, sa syntaxe vise à être lisible et visuellement épurée. Sans espérer une compatibilité complète, on pourra réaliser – et traduire – les algorithmes simples du niveau lycée.

Communiquer et partager

Enfin, il est possible d'un simple clic d'enregistrer un algorithme sur le site Proglab. L'algorithme peut alors très facilement être communiqué à d'autres personnes par n'importe quel canal (mail, chat, page web, document, ou même des messages courts du type SMS ou Twitter). Il suffit de copier le lien correspondant. Voici par exemple l'algorithme d'Euclide en Algobox : <http://proglab.fr/egs576>, et la courbe fractale de Von Koch : <http://proglab.fr/kyu931>. On peut ainsi imaginer des applications pédagogiques variées, depuis la simple mise en ligne d'un algorithme, jusqu'à des exercices à renvoyer par email.

L'histoire ne s'arrête pas là, et le site continue à évoluer. Les mécanismes d'aide et d'assistance sont amenés à se perfectionner. Des fonctionnalités supplémentaires pourraient apparaître pour ouvrir de nouvelles possibilités algorithmiques et pédagogiques. Avec toujours le même but en tête : rendre l'algorithmique plus accessible.

(7) http://fr.wikipedia.org/wiki/Virgule_flottante

(8) [http://fr.wikipedia.org/wiki/Python_\(langage\)](http://fr.wikipedia.org/wiki/Python_(langage))