

Les secondes passent au code

Fabien Cayla

Cet article est la synthèse d'un travail conduit en commun par trois enseignants du lycée Victor Hugo de Lunel, Fabien Cayla, Christophe Dellac et Gwenaëlle Lavoine (professeure stagiaire), lors de l'année 2013-2014. Les annexes évoquées contiennent des exemples d'exercices donnés à l'occasion de ce cours. Elles sont téléchargeables sur le site de PLOT.

Notre point de vue

L'une des difficultés de l'enseignement de l'algorithmique en classe de seconde consiste à trouver un travail abordable et cependant capable de mobiliser l'intérêt de l'élève. Ainsi, le type d'algorithme qui a pour données la longueur et la largeur d'un rectangle, et en sortie l'aire du rectangle est accessible, mais ne nous semble pas très intéressant pour un élève. D'ailleurs, il pourrait se traduire dans le supérieur par l'algorithme¹ suivant :

Données : Deux nombres L et l représentant la longueur et la largeur d'un rectangle.

Traitement :

Résultat : $L \times l$.

L'on voit en écrivant cet algorithme sous cette forme à quel point il est « creux ». Selon nous, il faut travailler sur des algorithmes différents qui posent de réels problèmes et qui soulèvent ainsi la curiosité des élèves. Pour pouvoir étudier de tels algorithmes, nous avons choisi d'enseigner les structures répétitives (boucles) et alternatives (si...alors) dès la seconde. Les objectifs du programme officiel d'algorithmique concernant les 3 années du lycée, nous avons décidé d'aborder dès la seconde :

- les affectations (savoir analyser, créer et modifier un algorithme comportant des affectations de variables) ;

- les structures répétitives et alternatives (essentiellement savoir analyser un algorithme comportant des structures répétitives ou alternatives).

Et de réserver à la première et à la terminale la création et la modification d'un algorithme comportant des structures répétitives ou alternatives.

Quelques pistes pour travailler sur les affectations

Des algorithmes intéressants

En dehors des algorithmes classiques de calcul de coordonnées du milieu de deux points ou de distance entre deux points, nous avons fait travailler les élèves sur les algorithmes suivants qui nécessitent (au moins au début) de dresser dans un tableau une trace de leur exécution :

- algorithme d'échange de valeurs ;
- algorithme de calcul de termes.

Comme, par exemple :

Entrée : deux nombres a et b que l'utilisateur choisit.

Traitement :

- * a prend la valeur a + 1
- * b prend la valeur b + 1
- * a prend la valeur a + b

Sortie : les nombres a et b

(voir annexe 1 sur le site de PLOT).

Des programmes faux

Nous avons proposé aux élèves des petits programmes écrits avec *AlgoBox* qui comportaient une ou plusieurs erreurs.

¹ Dans les documents ressources du lycée, les algorithmes sont écrits sous la forme suivante : variables, initialisation des variables, traitement, résultat. Cette forme n'est que très peu (pas du tout ?) utilisée dans le supérieur, où les algorithmes sont écrits sous la forme données/résultat.

Ces dernières pouvaient soit empêcher le programme de fonctionner, soit donner un résultat faux. Les élèves devaient trouver la raison du dysfonctionnement et corriger le programme.

Comme par exemple :

```

▼ VARIABLES
  |
  | -x EST_DU_TYPE NOMBRE
  | -y EST_DU_TYPE NOMBRE
▼ DEBUT_ALGORITHME
  |
  | -y PREND_LA_VALEUR 3*x*x-7*x+4
  | -LIRE x
  | -AFFICHER "La valeur de f("
  | -AFFICHER x
  | -AFFICHER ") est "
  | -AFFICHER y
  |
  | -FIN_ALGORITHME
    
```

Ce programme, écrit en *AlgoBox*, calcule l'image d'un nombre par la fonction $f(x) = 3x^2 - 7x + 4$.

- Expliquer pourquoi il ne fonctionne pas.
- Le corriger.
- Trouver un autre programme réalisant la même tâche et qui n'utilise qu'une seule variable.

L'erreur n'est pas si facile à détecter pour les élèves car, dans leur calculatrice, c'est en quelque sorte ainsi qu'on procède : on enregistre l'expression de la fonction, puis on donne des valeurs à x dans un tableau de valeurs pour calculer leurs images.

Sur les structures alternatives et répétitives

La découverte

Nous avons tout d'abord pensé à travailler avec le logiciel *Scratch*, qui est cité dans le document ressource « algorithmique ». Cela aurait permis de découvrir ces différentes structures en se dégageant des difficultés de syntaxe. Par manque du temps nécessaire pour concevoir des fiches de travail, nous repoussons cette séance jusqu'au moment où nous avons trouvé l'initiative « Hour of Code »², qui utilise ce même concept d'algorithme gra-

phique. Ce site web est très riche et très bien fait : tout est mis en place pour apprendre aux élèves à utiliser ce type de structure. Nous avons lancé les élèves sur la page du site <http://csedweek.org/learn> qui arrive ensuite à la page <http://learn.code.org/hoc/1>, constituée de petites énigmes.

Par exemple, voici l'énigme 11 :



Le petit oiseau rouge « angry bird » doit atteindre le cochon vert à l'aide de 4 « blocs » parmi les blocs suivants (un même bloc peut être utilisé plusieurs fois) :



Et la solution :



Essayer c'est l'adopter ! En une heure les élèves apprennent à utiliser des structures alternatives et répétitives et en redemandent ! Il faut ensuite relier cela au formalisme des algorithmes de seconde ; mais les concepts de structure répétitive et

² « Hour of Code » est une initiative américaine qui a pour but d'inciter les élèves à faire de la programmation. Dans le cadre de cette initiative de nombreuses personnalités (B. Obama, B. Gates, etc) interviennent pour inciter les élèves à apprendre à faire de l'informatique.

alternative ayant été compris, les élèves sont déjà capables de lire et d'effectuer « à la main » un algorithme utilisant ces structures (annexe 2 sur le site, PLOT).

Des suites

Comment cela des suites ? Certes dans les programmes de première et de terminale elles sont largement reliées aux algorithmes mais elles ne figurent pas au programme de seconde ! Eh bien cela ne pose aucun souci, tant que l'on n'emploie pas le terme « suite ». Voici le type de sujet que nous avons pu proposer :

Au 1^{er} janvier 2014, Caroline propose à ses parents qu'ils lui donnent de l'argent de poche selon le principe suivant : la première semaine elle reçoit 0,05 euros et, chaque semaine, elle reçoit le double de ce qu'elle a reçu la semaine précédente. »

Pour répondre à la question « Au bout de combien de semaines Caroline touchera-t-elle plus de 20 euros par semaine ? », on va utiliser l'algorithme suivant :

Initialisation : s prend la valeur 0,05

n prend la valeur 1

Traitement : Tant que $s < 20$

Début du tant que :

n prend la valeur $n+1$

s prend la valeur $s \times 2$

Fin du tant que

Résultat : afficher n

Les élèves devaient alors compléter un tableau représentant la trace de cet algorithme et répondre au problème posé. Cet exercice a été posé lors d'un devoir de synthèse en avril à quatre classes de seconde et a été réussi par la quasi-totalité des élèves.

Nous avons aussi évoqué des suites et des algorithmes avec un travail sur les pourcentages (évolution de la population d'une ville année par année, (annexe 3 sur le site, PLOT).

Des figures à tracer

Les essais réalisés avec *AlgoBox* sur les tracés de figures n'ont pas été satisfaisants. Finalement, peu d'élèves sont parvenus à programmer des tracés avec ce logiciel. Toutefois, en regardant un peu mieux le site de « Hour of Code », nous avons trouvé des idées intéressantes... développées à la base pour l'école primaire, mais personne n'est obligé de le dire aux lycéens !

L'année prochaine nous essaierons de faire travailler nos lycéens sur des pages comme celle-ci :

<http://learn.code.org/s/1/level/24> (une dizaine de tracés à faire, très progressif).

Conclusion

Les bénéfices de cette année : presque tous les élèves ont compris comment analyser un algorithme.

Les limites : ces activités (notamment pour les boucles) sont difficiles à intégrer dans le cours et beaucoup de séances restent un peu à part...

Nous avons finalement choisi d'enseigner l'algorithmique cette année avec plusieurs langages de programmation : un langage graphique (« Hour of Code »), un langage proche de l'algorithme (*AlgoBox*) et le langage TI (non développé ici). Chacun des langages choisis correspondait à un besoin précis. On aurait pu croire que cela perturberait les élèves mais finalement ce n'est pas ce que nous avons constaté. Ces différents langages leur ont-ils permis de concevoir le concept même d'algorithme ? Peut-être mais c'est difficile de l'affirmer ; par contre c'est la première année que nos élèves semblent assez à l'aise avec l'algorithmique.

L'initiative « Hour of Code » nous a bien aidés, et au moment où l'on parle de mettre du code à l'école primaire ou au collège, certains collègues pourraient être intéressés par les idées que l'on trouve sur ce site.