

Algorithmique et programmation : Présentation des fiches d'initiation

Tout d'abord, il faut préciser que certaines de ces fiches n'ont été élaborées qu'après avoir pris la mesure réelle des difficultés des élèves (et elles n'ont donc pas toutes été testées en classe).

Tous les manuels proposent des initiations à l'algorithmique, mais au vu des difficultés rencontrées par les élèves dans ce domaine nouveau pour eux, il semble que ces initiations ne soient pas toujours assez progressives.

Les fiches d'initiation proposées ici tentent d'introduire progressivement l'algorithmique, en insistant sur ce qui semble le plus problématique pour les élèves, en particulier la notion de variable. Ces fiches, dans l'ensemble, sont indépendantes les unes des autres et généralement une fiche peut être faite en classe sans avoir fait les précédentes (exception: dans les fiches 3 et 6, on fait référence à la fiche 2)

Fiche 0 "Algorithmique : de quoi s'agit-il ?"

C'est une première approche qui met l'accent sur la notion d'entrée et de sortie en faisant le lien avec des activités de la vie courante et des activités mathématiques connues des élèves (médiatrice d'un segment, moyenne de trois nombres).

Cette fiche n'a pas été testée en classe en l'état, elle est la dernière mouture de plusieurs fiches d'initiation testées puis modifiées.

Fiche 1 "Pour bien démarrer : les mémoires de la calculatrice" (ne nécessite pas d'avoir fait la fiche précédente)

Cette fiche cherche à familiariser les élèves avec les mémoires de la calculatrice. Car pour introduire les variables, il est intéressant de les imager par des "boîtes-mémoires" (voir fiches suivantes), d'ailleurs plusieurs manuels ont pris ce parti aussi.

Cette fiche a été testée dans trois classes, en module. En une heure, les élèves ont le temps de traiter les parties "Mémoire courte" et "Mémoires longues" et de faire le premier exercice (certains ont le temps de finir la feuille d'exercices)

On s'est rendu compte que cette fiche nécessite une certaine maîtrise de la calculatrice : calculs avec puissance et quotient (possibilité d'oubli de parenthèses avec la TI). Dans une des classes, les élèves avaient auparavant déjà utilisé la touche *rép* ou *ans* de leur calculatrice et semblent avoir éprouvé moins de difficulté que les autres élèves pour traiter la partie "Mémoire courte".

Fiche 2 "Des suites d'instructions et des robots" et Fiche 3 "Des suites d'instructions et des boîtes-mémoires"

Certains manuels proposent, comme exercices préliminaires, des suites d'instructions aisément compréhensibles par les élèves : choisir un nombre – le mettre au carré – ajouter 2 – etc .

Puis ils proposent, sans réellement l'expliquer, une transposition de ces suites d'instructions dans un langage plus ou moins codé.

Or la marche semble haute entre les deux étapes et ces manuels semblent minimiser la difficulté de certaines instructions :

- ❖ les instructions du type : « b prend la valeur $a - 4$ » (appelées aussi affectations) comportent des sous-entendus :
il faut comprendre qu'on soustrait 4 à la valeur qui est dans a et qu'on place le résultat dans b
Cela nécessite de maîtriser la notion de variable et c'est en cela que l'utilisation des boîtes-mémoires est intéressante.

- ❖ l'instruction « Entrer a » est également plus complexe qu'il n'y paraît.

En effet, il faut réaliser que cette instruction permet trois choses :

- demander à l'utilisateur d'entrer un nombre,
- définir le nom de la variable où va être rangée la réponse de l'utilisateur,
- stocker la valeur entrée par l'utilisateur

La fiche 2 a été testée en demi-classe, en trente minutes environ. Elle met aussi en jeu des notions mathématiques telles que: décomposer un calcul, montrer une égalité (avec discussion autour de la valeur des exemples).

La fiche 3 a été testée en classe entière et a nécessité, selon les élèves, entre 15 et 25 minutes.

Fiche 4 " Des suites d'instructions et des variables" Fiche 5 " Trouver le but d'un algorithme, écrire ou modifier un algorithme"

Dans la fiche 4, on introduit le terme « variable ». Il s'agit aussi ici d'appliquer un algorithme écrit sous forme contrainte.

L'exercice 3 est là pour réaliser que des algorithmes d'écritures un peu différentes peuvent avoir le même but.

En particulier, le nombre de variables peut changer.

Les fiches 4 et 5 ont été données dans une classe et cela a pris environ 45 minutes pour les deux.

Fiche 6 " Des suites d'instructions et un logiciel"

On programme sous algobox un algorithme déjà écrit sous deux formes. Ce qui nous donne trois niveaux de langages pour un algorithme.

Après avoir écrit l'algorithme sous algobox, on leur demande d'en vérifier la cohérence avec un exemple.

Cette fiche n'a pas été testée en classe. Pour certains élèves, il faudrait sûrement prévoir d'autres exercices en plus.

Fiche 7 " L'instruction Si ... alors ..." Cette fiche n'a pas été testée.

Fiche 8 " Choisir les variables"

Certaines instructions sont réellement très difficiles à comprendre pour nos élèves : on les rencontre souvent à l'occasion des boucles mais on peut commencer à soulever le problème auparavant.

Il s'agit d'affectations du type : « N prend la valeur N + 1 » ou encore « S prend la valeur S + K »

Ici, il faut bien entendu comprendre que la *nouvelle valeur* de N s'obtient en ajoutant 1 à l'*ancienne valeur* de N.

Le but de la fiche 8 n'est pas d'apprendre à réduire systématiquement le nombre de variables d'un algorithme.

Par exemple, il ne nous viendrait pas à l'idée de désigner par une même lettre un nombre et son image par une fonction, cela risquerait d'être perturbant pour nos élèves.

Mais ces exercices sont là pour aider à se familiariser avec l'idée qu'une même variable peut prendre différentes valeurs selon l'état de ce qu'elle représente : valeur initiale/ valeur finale et parfois même il y aura des valeurs intermédiaires ...

L'idée est bien d'utiliser une même variable pour des paramètres de même nature (et derrière il y a la notion de suite de nombres).

Ainsi, utiliser une seule variable pour désigner un prix (initial) qui subit une variation et le prix final qui en résulte peut parfois être judicieux.

Fiche 9 " Boucle pour : initiation et applications"

Les situations 1 et 2 mettent en évidence le fait qu'il serait bien pratique de disposer d'outils évitant d'écrire plusieurs fois la même instruction (ou évitant d'écrire plusieurs fois des instructions, qui sans être identiques, se ressemblent beaucoup.)

L'instruction REPETER n FOIS n'existe que dans peu de langages informatiques mais elle a semblé intéressante ici : utiliser d'emblée POUR i ALLANT DE 1 à N semblait poser plus de difficultés dans la mesure où dans ce cas la variable i sert uniquement de compteur et n'est pas employée dans les instructions de la boucle (ce qui est troublant au départ).

Le fait que les deux situations peuvent se traiter avec la boucle pour est mis en évidence dans l'exercice 3 de la fiche applications.

Ces fiches ont été testées dans trois classes et dans deux d'entre elles le bilan est plutôt positif.

Fiche 10 " Tant que : initiation"

Cette fiche a été testée presque dans son intégralité, l'exercice 1 a été un peu modifié pour plus de clarté. Elle est longue : elle n'est pas conçue pour une séance d'une heure. Elle nécessite l'usage de l'ordinateur pour programmer sous algobox. Ce serait bien que les élèves aient le temps de programmer l'algorithme de l'exercice 3 **après l'avoir fait tourner à la main** : ils ont du mal à réaliser qu'ils doivent appliquer l'instruction plusieurs fois.

La difficulté du tant que est liée en grande partie à la condition d'arrêt pas toujours facile à interpréter et surtout plus tard à écrire, avec le risque d'un algorithme qui tourne indéfiniment.