COMMENT PEUT-ON ABORDER LES PROBLÈMES DE PROGRAMMATION DANS LA NOUVELLE OPTION INFORMATIQUE DE SECONDE.

Eric GREFF

INTRODUCTION

Autrefois, lors de la première mouture de « l'option informatique », l'accent était particulièrement mis sur l'apprentissage de la programmation à partir de la connaissance de l'algorithmique et de l'utilisation d'un langage informatique évolué. Les élèves se confrontaient aux rudiments du Pascal et créaient des programmes permettant, grâce à l'habile déplacement d'un pointeur dans un tableau T[i,j], de calculer la moyenne d'un élève de la classe...

Outre cette tendre ironie (y compris à l'égard de moi-même), force est d'admettre que le lycéen abordait alors des notions difficiles de programmation dans un langage structuré qu'il aurait eu bien du mal à appréhender seul. De plus, les structures mentales que développe ce type d'activité offraient, probablement, des « bases de pensée » irremplaçables. Malheureusement, les applications que l'on pouvait proposer ressemblaient trop souvent à des exercices d'école ou aboutissaient à des programmes finalement peu utilisables...

Puis vint le temps révolu des ateliers de pratique informatique dans lesquels seule l'utilisation de logiciels (professionnels ou non) était mise en avant.

L'année 95/96 marque le début d'une nouvelle option informatique en classe de seconde (B.O. n°18 du 4 Mai 1995). Son ambition semble être de s'appuyer sur des environnements et des logiciels courants et usités pour découvrir les notions informatiques principales. Il ne s'agit plus désormais d'apprendre un langage informatique particulier mais il convient cependant de savoir que votre tableur ou votre jeu préféré à lui même été programmé. On tentera alors de ne plus utiliser le produit

informatique sans chercher à « voir » et à comprendre les notions essentielles qui ont présidé à sa création. Il s'agit, entre autres choses, de faire comprendre que toute personne désirant créer un produit grâce à l'ordinateur est confronté au problème du « faire faire », donc à des problèmes relevant de la programmation.

La question qui se pose alors est de savoir quelles sont les notions informatiques que l'on peut réellement aborder à partir des systèmes d'exploitation et des logiciels courants. Au-delà de « faire faire », il s'agira d'expliquer comment l'ordinateur fait ce qu'il fait et par conséquent, ce qu'il ne peut pas faire et pourquoi. Les exemples qui suivent ne prétendent, en aucun cas, fournir une liste exhaustive des exercices possibles ni constituer un « cours » de l'option informatique en Seconde mais servent uniquement à illustrer le propos. Il s'agit simplement d'observer quelles notions informatiques on peut étudier, d'une part à travers l'utilisation de logiciels professionnels (en particulier les tableurs) et, d'autre part, en s'interrogeant sur la programmation sous-jacente à ces logiciels.

Rappelons bien que, dans le premier cas, il s'agit de se placer au niveau « utilisateur » du logiciel et d'étudier les aspects liés à la programmation uniquement à travers leur fonctions de base sans aborder les macros.

Nous nous attacherons tout particulièrement à quatre notions importantes de l'algorithmique, à savoir : les variables, la séquentialité, l'alternative et l'itération.

VARIABLES

La notion de variable informatique est didactiquement ardue. Il est difficile de l'isoler des notions de saisie, d'affectation et d'affichage. Elle est également liée aux concepts de type et de valeur et se heurte à la difficulté de nommer précisément les objets sur lesquels on travaille.

Du point de vue de l'utilisateur du logiciel

ler temps : la possibilité offerte par les tableurs modernes d'utiliser la souris pour sélectionner ou déplacer les cellules désirées permet encore de mieux focaliser son attention sur « ce que l'on veut faire faire » (ex : ajouter le contenu de cette cellule au contenu de celle-ci) en oubliant, momentanément, la syntaxe associée. Il suffit donc, à ce stade, de se con

centrer sur le but à atteindre sans se soucier outre mesure de la manière d'y parvenir.

2ème temps : on s'intéresse à la ligne de commande que génèrent les déplacements de la souris et les opérations que l'on a saisies. Chaque cellule est alors considérée comme une variable dont le nom est constitué du numéro de ligne et de colonne correspondant à sa position. Son type, s'il n'est pas pré-défini, peut être du texte, du numérique ou du logique et est fixé lors de l'affectation. Cependant ce typage est faible, il n'est pas définitif et peut être remis en cause lors d'une nouvelle affectation. Il faudra néanmoins tenir compte du type d'une cellule lors de son utilisation dans une combinaison ou dans une fonction.

3ème temps : on peut enfin aller plus loin dans ce travail en « nommant » une cellule par un identifiant significatif pour travailler ensuite sur celui-ci en dehors de sa valeur associée. Ceci permet alors d'introduire, entre autre, la notion d'adresse et les notions de références relatives et absolues.

Questions sur la programmation du logiciel

L'observation des lignes de commandes crées par le tableur permet d'appréhender la manière dont il fonctionne de manière interne. Le concept de « format » permet, par exemple, d'aborder celui de représentation-machine et de montrer comment une même valeur peut être exprimée de façons différentes à l'écran. De manière plus générale, un tel support associé à des exercices adaptés permet d'appréhender précisément la manière dont les données sont représentées en mémoire et se présentent à l'utilisateur. Ils autorisent conjointement la compréhension des possibilités et des limites des logiciels utilisés.

Conclusion

On peut étudier, grâce notamment au tableur (ou à un gestionnaire de bases de données), la notion de variable et celles d'identifiant, de type et de valeur qui lui sont associées. Le tableur constitue à ce titre un excellent outil. La manière de procéder évoquée (souris puis modification de la ligne de commande) constitue également une excellente aide à la programmation puisque l'on peut ensuite modifier la ligne de commande ainsi créée. Le tableur permet donc d'aborder facilement la notion de variable et d'effectuer les exercices de familiarisation qui se présentaient autrefois sous la forme A←3 ; B←A ; C←A+B.

LES STRUCTURES DE CONTRÔLE

Bien qu'elles ne soient pas envisagées dans le programme de Seconde, ce chapitre permet de montrer que, si les structures de contrôles classiques sont intrinsèquement présentes dans les logiciels professionnels ou dans les systèmes d'exploitations que nous avons abordés jusqu'à présent, elles ne pourront pas être facilement étudiées, à travers les fonctionnalités des logiciels, dans la suite de la progression en Première, par exemple).

« Faire faire », c'est aussi « faire refaire » ou « ne faire que si... ». Les notions essentielles de l'algorithmique et de la programmation structurée ne peuvent être qu'effleurées à travers ces supports originaux qui marqueront là leurs limites...

L'ALTERNATIVE

La difficulté didactique principale de l'alternative est constituée de la rupture de séquentialité qu'elle génère. Une structure algorithmique de type « Si... Alors... Sinon » éclate le programme en 2 voies qui se séparent pour se rejoindre éventuellement, plus tard, lorsque l'une ou l'autre aura été empruntée.

Du point de vue de l'utilisateur du logiciel

Sur ma feuille de calcul, j'aimerais que les nombres négatifs soient inscrits en rouge. Autrement dit, si certaines conditions sont remplies, je voudrais que certains traitements soient effectués. Le logiciel met à ma disposition une fonction linéaire de type « =SI(Condition ; Valeur1 ; (sinon) Valeur2) ». Il s'agit bien là d'une approche fonctionnelle et non impérative de l'alternative qui renvoie une valeur et ne fait pas exécuter d'actions. Ceci demeure donc de type linéaire et ne permet pas de faire émerger, sans explicitation particulière, le caractère « dimension 2 » caractéristique de l'alternative.

Des logiciels comme le tableur ou le gestionnaire de bases de données requièrent l'utilisation d'une syntaxe spécifique (ce qui est également un élément de programmation) pour approcher l'alternative. Si la cellule « Bl9 » est supérieure à zéro, alors j'écris « solde positif » dans la case de mon choix. Si le contenu du champ « Code postal » commence par « 19 », alors il faut afficher les contenus des champs « nom » et « prénom » de l'enregistrement correspondant.

Questions sur la programmation du logiciel

On pourra tenter de repérer des instructions de type « alternative » se dissimulant dans certains programmes afin de tenter d'expliciter les conditions et les actions qui leur sont liées. On s'interrogera, par exemple, sur la présence à l'écran d'un message du type « fichier verrouillé, ne pourra pas être modifié » ou sur la causalité d'apparition d'un message d'erreur dans la cellule d'un tableau. Ces exercices permettront de mettre en évidence l'existence et le fonctionnement de nombreuses structures alternatives sous-jacentes aux programmes informatiques.

Conclusion

L'utilisation de telles structures permet essentiellement d'expliciter sous quelle forme doivent se présenter les conditions, d'introduire les expressions booléennes, de réfléchir à ce que le logiciel peut exécuter de manière conditionnelle, bref de développer une réflexion approfondie sur la notion de fonctions conditionnelles mais pas vraiment sur l'alternative.

L'ITÉRATION

Du point de vue de l'utilisateur du logiciel

La répétition indicée est souvent liée aux tableaux dans la programmation classique. On la trouve donc de manière claire dans les feuilles de calcul. On peut, en effet, itérer une formule de calcul sur le nombre de cases choisi. Les dernières versions d'Excel permettent notamment « d'étendre » une série sur le nombre de cellules désiré. Lorsque l'on sélectionne une « zone » sur laquelle on applique un format, on ne fait rien d'autre, intrinsèquement, que répéter l'application de ce format le nombre de fois nécessaire à couvrir la zone. Dans une base de données, on peut rechercher l'existence d'un nom parmi les 354 fiches saisies. On répétera alors 354 fois l'action consistant à comparer le nom saisi avec le nom de la requête. Ce faisant, j'explique le fonctionnement interne du logiciel, je ne crée pas, personnellement, de répétition.

Questions sur la programmation du logiciel

La répétition conditionnelle peut également être effleurée à l'aide des logiciels professionnels courants. En effet, lorsque vous recherchez un mot dans votre texte, le programme vous propose, après la première occurrence de trouver le « suivant ». Cette itération se poursuivra jusqu'à

ce que l'expression recherchée n'apparaisse plus. On peut également appréhender cette structure de contrôle en explorant une base de données. Par rapport à l'exemple précédent concernant la recherche d'un nom précis, on peut décider d'arrêter le défilement dès que l'on a trouvé la première occurrence et négliger les problèmes d'homonymie.

Cependant et avec les mêmes réserves, on pourra s'interroger sur l'utilisation interne que le programme fait de l'itération. Est-elle indicée ? conditionnelle ? Sur quelle condition s'appuie-t-elle ? Comment travaille-t-elle ?

Conclusion

Les exemples ci-dessus mettent bien aussi en évidence les limites flagrantes des logiciels (hors macro-commandes) en ce qui concerne les structures itératives. En effet, il n'existe pas d'itération directe et l'on ne peut aborder celle-ci qu'à travers l'explicitation des fonctionnalités internes au logiciel, ce qui est didactiquement discutable.

CONCLUSION

Associer l'apprentissage de l'informatique à l'utilisation de logiciels ou de systèmes d'exploitations courants présente comme principal intérêt de ne pas « déconnecter » cet enseignement de la réalité commune et professionnelle. On a trop reproché, à juste titre ou non, mais l'heure de ce débat est dépassée, à la première option informatique de ne pratiquer que des « exercices de style » ne correspondant pas aux besoins de l'utilisateur courant de l'ordinateur. L'ambition d'alors était-elle réellement de former de futurs programmeurs ou plus simplement d'appréhender, « en faisant soi-même », les notions essentielles propres à l'informatique ?

Dans ce dernier cas, cette nouvelle option inaugurée en classe de seconde à la rentrée 95/96 poursuit, a priori, le même but en utilisant comme nouveau moyen le « faire faire ». Elle peut donc être considérée, si l'on est très vigilant et rigoureux, comme la réconciliation entre l'apprentissage de l'utilisation de produits courants existant sur le marché et des concepts fondamentaux sous-jacents qu'il sera essentiel de ne pas négliger au risque de revenir malencontreusement aux « ateliers de pratique ».

Cependant, l'utilisation de logiciels ne permet pas d'aborder convenablement tous les concepts profonds de la programmation et il faudra

bien, tôt ou tard, trouver d'autres moyens pour y parvenir. L'utilisation des macro-commandes devra alors être abordée, si l'on désire poursuivre dans cette même voie. Est-ce là le meilleur choix ? C'est un autre débat.

Il est donc simplement, pour l'instant, question de ne pas « informatiser idiot ». Gageons que nos nouveaux « informaticiens », par leur connaissance des fonctionnements externes et internes de leur ordinateur et de leurs logiciels, sauront en tirer un meilleur profit... en attendant ...

Éric GREFF