

Enseigner l'informatique aujourd'hui

Beaucoup de jeunes et de moins jeunes s'intéressent à l'informatique et parfois achètent un ordinateur avant même de savoir ce qu'il pourront en faire. Si le matériel est aujourd'hui accessible au plus grand nombre, la pédagogie de l'informatique n'en est qu'à ses balbutiements. Cette constatation nous a amenés à élaborer une méthode orientée vers l'éducation populaire et la formation permanente.

LANGAGES, MATÉRIELS ET LOGICIELS.

Sans méconnaître les indéniables avantages de langages à finalité didactique comme le LOGO, structuré comme le BASIC ou « francisé » comme le LSE, nous avons opté pour une démarche essentiellement pragmatique.

Pour que l'enseignement soit accessible au plus grand nombre, il faut que les exercices d'application puissent se faire sur tous les micro-ordinateurs existant sur le marché et notamment les « bas de gamme » et ordinateurs de poche.

Plusieurs années d'enseignement de l'informatique et d'animation de clubs nous ont conduits à perfectionner un langage, le « basic minimum », constitué de moins de vingt mots clés et muni d'une syntaxe simplifiée.

Précisons immédiatement que la portabilité de ce langage est totale, c'est-à-dire qu'un programme écrit dans celui-ci est utilisable sur tous les micro-ordinateurs sans pratiquement aucune transcription. Cette portabilité constitue l'un des deux objectifs que nous nous sommes fixés, l'autre but étant de limiter au minimum le temps consacré à l'apprentissage initial du langage au profit de l'algorithmique et surtout à la pratique.

Nous proposons donc une série de programmes constituant tout à la fois des exercices et des « modèles réduits » de logiciels professionnels.

Nous pensons d'ailleurs que pour captiver l'utilisateur, il faut que les exemples proposés n'apparaissent pas comme trop rébarbatifs au premier abord. Ce qui conduit à séparer clairement les centres d'intérêts : par exemple, une orientation littéraire axée sur la poésie, les langues, la grammaire, les tris et le traitement de texte, une orientation calcul axée sur les méthodes de calcul numérique, la comptabilité, la gestion, l'aide à la décision et une orientation scolaire englobant les jeux, le graphisme et l'enseignement assisté par ordinateur.

Première étape de l'enseignement :

Apprendre à écrire, structurer et utiliser un programme.

Dès le premier contact avec l'informatique, il faut savoir découvrir l'ordinateur : c'est-à-dire trouver rapidement dans la documentation souvent surabondante les modes d'emploi essentiels pour introduire, lister, exécuter, interrompre et effacer un programme. Une fois ce premier contact établi, on pourra alors s'intéresser au langage proprement dit. On étudiera notamment l'utilisation des mots clés essentiels (PRINT, INPUT, GOTO, GOSUB, LET, RETURN, IF-THEN, DIM, ABS, INT, REM, END).

Il faut remarquer tout de suite que notre « langage » est un véritable langage de programmation. Bien qu'il n'utilise qu'une douzaine d'instructions, il est suffisant pour construire des structures répétitives, alternatives et élémentaires (simples avec ou sans routines).

Malgré les contraintes découlant de l'utilisation d'un tel langage, nous

avons pu écrire un grand nombre de programmes courts pouvant se classer dans les orientations déjà citées et qui constituent en outre un fond utilisable ultérieurement comme logithèque de sous-programmes.

On fera attention de ne pas passer trop vite cette étape. Il sera utile de reprendre les exercices proposés pour les perfectionner en fonction de son ordinateur personnel ou d'adapter les programmes trouvés dans les livres d'informatique.

Deuxième étape de l'enseignement : Explorer l'ordinateur, dépister les erreurs :

Il est inutile de parler de « bogues » et de messages d'erreur avant d'en avoir fait l'expérience personnelle. Ce n'est qu'après avoir suffisamment programmé qu'on manifeste un intérêt pour ce que cache l'ordinateur et qu'on part à la découverte du « répertoire d'injures » continues en mémoire morte, à l'aide de petits programmes volontairement erronés.

Etablir un lexique complet des messages d'erreur de l'ordinateur est un exercice d'autant plus utile que chaque modèle à son dialecte (et que les notices souvent traduites de l'anglais ou du japonais ne sont pas toujours très claires pour un débutant). Une question qui est négligée dans la plupart des livres d'initiation à l'informatique, est celle des erreurs non détectées par la machine et conduisant celle-ci à donner un résultat... mais un résultat faux (ce sont pourtant les erreurs les plus difficiles à trouver).

Les techniques d'exécution manuelle « d'instruction-espion » et de vidage de la mémoire devraient être introduites à ce niveau.

On n'insistera jamais assez sur la nécessité de ne pas écrire des programmes qu'on ne soit pas capable de « débrouiller » soi-même. Nous pensons que malgré son caractère fastidieux, la technique de l'exécution manuelle doit être systématiquement pratiquée avec des débutants ; le rôle de l'enseignant est ici indispensable.

Troisième étape : Découverte du langage BASIC.

Si les deux étapes précédentes ont été parcourues sans trop de hâte, il n'y a alors plus aucune raison d'interdire l'usage du langage complet.

Cette étude peut se faire individuellement à l'aide de la documentation de l'ordinateur et de livres spéciali-

sés mais auparavant, il serait bon de traiter des questions suivantes : sauvegarde et appel des programmes, manipulation des chaînes de caractères, instruction raccourcissant un programme, gestion de fichier, fonctions mathématiques courantes dans la mesure évidemment où les instructions appropriées sont disponibles sur le matériel utilisé. Le tout doit continuer à être illustré par des programmes nombreux et attrayants. (La difficulté réside ici dans le fait que les programmes doivent avoir une portabilité maximum ce qui n'est pas chose facile vu la grande diversité des basics).

Il peut être intéressant ici de reprendre les programmes vus au niveau de la première étape de l'enseignement pour en faire de véritables logiciels autonomes et performants.

Retombées culturelles et pédagogiques.

Il nous semble important que l'initiation à l'informatique ne soit pas conçue comme une formation encyclopédique d'utilisation d'une simple machine mais un apprentissage de quelques méthodes générales en informatique et indirectement une façon de compléter l'enseignement reçu depuis l'école primaire dans les domaines les plus variés : il est bien connu que les lacunes scolaires constituent un handicap socio-culturel que nous adultes avons beaucoup de difficultés à combler.

Les auteurs de livres et les animateurs de clubs informatiques doivent avoir une claire conscience de leur mission dans ce domaine. Aussi, les programmes proposés à titre d'exercice devront avoir été choisis et préparés soigneusement pour qu'ils n'apportent jamais de connaissances générales erronées et si possible aient un contenu didactique. Les mots d'origine anglaise devront être expliqués en signalant leur étymologie.

De plus, on n'insistera jamais assez sur la nécessité de préparer ses programmes à l'avance et non de les taper « au petit bonheur la chance » directement sur l'ordinateur. Ces programmes devront avoir été rédigés auparavant d'une façon claire, précise et lisible. Nous estimons que cette contrainte est pédagogiquement utile.

Enfin, nous pensons que dès le premier contact avec l'informatique, les apprentis informaticiens devront être mis clairement au courant du fait qu'ils ne seront pas capables de pratiquer utilement l'informatique du jour au lendemain, (contrairement à ce qu'affirment les médias). Cela en découragera peut être quelques-uns, mais la majorité sachant exactement ce qu'il en est n'en sera que mieux armée pour aborder la plus importante révolution technologique de notre temps.

Philippe BONBOIRE.
Cyril DEICHA.