

LES BASICS POUR PC : QUE CHOISIR ?

Jean-Louis MALANDAIN

La question paraîtra saugrenue à bien des collègues de mathématiques ou de sciences ; ils auront un petit soupir de soulagement en apprenant qu'elle est posée par un professeur de français, même pas par un linguiste !

Pour clarifier encore les choses, précisons que l'objectif recherché est de gérer des affichages à l'écran et d'introduire dans les textes des séquences limitées faisant appel à l'interactivité.

Il ne s'agit donc pas d'analyser ou de simuler les fonctionnements de la langue, ce que ferait peut-être mieux LISP ou PROLOG (encore que J.-P. BALPE...), ni de constituer des bases de données (encore que...), ni de réaliser des travaux de lexicographie (encore que...), ni de transformer un PC en MAC pour naviguer dans l'hypertexte.

Il s'agit, tout simplement, et en attendant que soient accessibles des systèmes auteurs ou des logiciels de présentation ou des outils d'infographie, de travailler avec ce qu'on a et de faire ce qu'on peut...

Alors se pose le problème du choix éventuel d'un Basic.

Le Gwbasic (ou Basica pour la marque IBM) est le produit de base, généralement vendu avec la machine. Sur le dernier modèle d'IBM, le PS/1, il est même "soudé" dans la machine.

Le Turbo-Basic de BORLAND, disponible au catalogue 1990-91 de la CAMIF pour 845 F. est le moins cher (mais 870 F. au catalogue 1990 de l'UGAP).

Le Quick-Basic de MICROSOFT coûte 865 F. à la CAMIF et 850 F. à l'UGAP (version 3 ou 4, non précisée).

Question : pourquoi passer du premier (gratuit) aux deux suivants ? Les avantages sont nombreux.

Le GWBASIC lit une séquence quelconque écrite selon une syntaxe spécifique et présente immédiatement sa réalisation à l'écran. On dit de GWBASIC que c'est un "interpréteur" ; il faut donc l'avoir sous la main pour exécuter un programme.

L'utilisateur éventuel d'un programme doit connaître et effectuer un certain nombre d'opérations préalables (souvent ignorées et source de rejet) : 1 installer son DOS, 2 appeler son Basic, 3 charger le programme, 4 lancer le programme.

Si tous les fichiers nécessaires sont sur le disque ou la disquette (DOS + BASIC + PROGRAMME), on peut se simplifier la tâche en tapant BASIC nomprog (pour nom du programme à exécuter) avant de valider (à moins d'écrire un fichier batch qui automatise ces opérations). La moitié des utilisateurs potentiels a déjà renoncé à comprendre ces bizarreries !

Les deux autres Basics, TB et QB, qui sont des "compilateurs", permettent de générer un programme directement exécutable sous DOS : il suffit de l'appeler par son nom et de valider. C'est quand même plus simple !

Prenons le cas d'un PC avec un seul lecteur de disquette, sans disque dur. Pourquoi riez-vous ? C'est cruel de se moquer des pauvres... surtout dans l'enseignement primaire et secondaire où la modestie s'impose.

Il faut trouver la disquette système, l'introduire dans le lecteur, le verrouiller puis allumer l'écran et l'ordinateur. Eventuellement modifier la date et valider. A l'apparition du signal d'attente A>, il faut enlever la disquette système et la remplacer par celle où se trouve le programme : son nom est généralement inscrit sur l'étiquette et il suffit de le taper puis de valider pour le voir s'exécuter.

L'autre avantage des nouveaux Basics concerne l'écriture du programme. On dispose de presque tous les avantages d'un traitement de texte, avec les fonctions de recherche/remplacement et la possibilité de définir des blocs pour les supprimer ou les déplacer. Selon qu'on est habitué à l'ergonomie de Wordstar (début et fin de bloc, effacement, déplacement, copie) ou à la technique du "couper/coller", on sera plus à l'aise dans TB ou dans QB. Mais les deux éditeurs apportent un confort réel.

Il est vrai qu'on peut toujours rédiger un programme à partir d'un traitement de texte puis le sauver en ASCII pour le faire exécuter par le GWBasic mais la numérotation des lignes restera un handicap à tout déplacement ou insertion d'un bloc ou d'un sous-programme. En QB ou TB, cette numérotation n'est plus indispensable : on peut la garder ou s'en passer selon les besoins. Les branchements (GOTO) ou les appels de sous-programmes (GOSUB) se font par désignation d'une étiquette.

A-t-on assez reproché au Basic les imbrications macaroniques de GOTO 1255 ou GOSUB 3010 rendant pratiquement impossibles les relectures et les modifications ultérieures par un tiers. Les adeptes de la programmation structurée criaient au scandale et n'avaient que mépris pour tous ceux qui se livraient à cette gymnastique... et ils étaient nombreux à le faire puisque c'était le seul langage disponible. Combien d'amateurs ont commencé ainsi et n'ont pas eu le temps ou le courage ou l'occasion de recommencer une initiation en Pascal ? C'est inavouable ! Ils peuvent désormais, pour moins de 1000 F. (quand même !) faire des programmes avec des simulations de procédures portant des noms évocateurs, de belles écritures indentées et des lignes sans numéros d'allure très pascalienne.

CHOIX:

```
LOCATE 10,1
PRINT "Voulez-vous une explication ou continuer ? (E/C)"
  C$=INPUT$(1)
  ON INSTR("EC",C$) GOTO EXPLICATION,SUITE
GOSUB MESSAGE
GOTO CHOIX
```

MESSAGE:

```
PRINT "Tapez E ou C pour choisir"
RETURN
```

Les inconvénients rencontrés sont peu nombreux mais réels : avec GWBasic, il est possible de travailler en mode direct (sans même numéroter les lignes) pour essayer une fonction où tester une réalisation sans perdre son écriture, par exemple voir ce que donne DRAW "BM183,45D20M-10,+13" puis revenir dans cette ligne pour effectuer des modifications avant de la conserver en lui attribuant un numéro. En TB ou QB, il faudra sortir de l'éditeur, donc cesser de voir les instructions pour les voir se réaliser.

L'obstacle majeur, commun à tous les logiciels actuels, est l'encombrement en mémoire vive ou de masse. Le GWbasic est un fichier de 70 Ko qui génère des programmes limités à 64 Ko. Le TB tient encore sur une disquette où il occupe 260 ko mais la dernière version de QB 4.50 est servie sur 5 disquettes et impose l'emploi d'un disque dur où elle occupe 2 Mo (avec une abondante documentation en ligne). Les fichiers générés sont souvent volumineux : 30 Ko de GWBasic deviennent 70 Ko ou plus une fois compilés. On aura parfois des difficultés à compiler

d'anciens programmes Basic trop lourds, surtout en QB, alors qu'en principe c'est une opération facile : tout programme écrit en GWBasic et sauvé en ASCII (option SAVE "nomprog",A) peut être appelé puis compilé en TB ou QB .

Pour les programmes sans complexité, les quelques différences de syntaxe entre GWB et les nouveaux Basics sont généralement bien signalées au moment des essais. On aura cependant intérêt à vérifier les mots interdits ou les incompatibilités : pour l'utilisateur venant de GWB, c'est le premier chapitre à lire ! Aussi bien TB que QB ont placé ces informations dans les annexes des documentations, comme si c'était une filiation douteuse...

Enfin, et c'est le plus grave, le TB de BORLAND, pourtant plus économique, moins encombrant, comporte des bogues particulièrement pénalisantes pour gérer des textes : non reconnaissance des caractères accentués et aberrations dans la gestion des DATA un peu lourds. L'éditeur BORLAND reconnaît ces défauts mais le logiciels n'a pas été corrigé ; il a été retiré du catalogue (façon élégante de régler le problème !). On peut toujours se servir de TB pour écrire et tester les programmes avec une seule disquette, avant d'aller les compiler sur la machine où est implanté le QB (la communication est bonne entre les deux marques dans le sens TB vers QB).

Décidément, les littéraires pauvres n'ont pas de chance. Ils ont souvent commencé avec un LSE gratuit qui est devenu cher et, surtout, n'a jamais amélioré son éditeur, à peu près aussi rigide qu'EDLIN. Ils ont pu trouver dans GWBasic un instrument simple pour gérer le texte à l'écran et TB aurait apporté un confort indéniable pour l'écriture et la diffusion de programmes compilés.

En principe, mais c'est tout nouveau, Borland a cédé son Turbo Basic à AB Club qui le diffuse sous le nom de Power Basic au prix de 1280 Fht... mais il faudrait s'assurer que les bogues ont disparu.

Il reste QB pour travailler en toute sécurité et confortablement, à condition d'avoir un disque dur pour éviter le jeu du grille-pain (appel des disquettes où se trouvent les différents fichiers secondaires).

BIZARRE !

La copie d'écran par programme, tellement simple en BASIC (il suffit de demander LCOPY) est devenue CALL INTERRUPT 5 en TB et est presque un secret d'Etat en QB (téléphoner à Microsoft).

La gestion des variables dans l'écriture de dessins ou de musiques, relativement claire en GWBasic, est devenue une gymnastique mystérieuse en TB et en QB.

La gestion de la hiérarchie des opérateurs arithmétiques (qui marque si bien la différence entre les littéraires et les autres dans les stages d'initiation !) donne lieu à des nuances intéressantes à observer :

- en GWB, $319 * 199 / 100 = 634.81$
 $319 / 100 * 199 = 634.81$
- en TB, on obtient respectivement -20.55 et 634.81
- en QB, on obtient respectivement "dépassement de capacité" et 634.81

En QB, l'opération de compilation passe par une étape intermédiaire qui génère un fichier "objet" dont la taille ne doit pas excéder 64 Ko, sans qu'on sache très bien ce qui va gonfler un fichier en GWBasic au-delà de cette limite.

En TB, si on n'a pas pris la précaution d'activer Ctrl-break ou de sauvegarder un programme avant de l'essayer, on peut tout perdre car certaines erreurs de syntaxe provoquent tout simplement le plantage de la machine...

En TB, le résultat de l'écriture PRINT INSTR(A\$,B\$) est parfois aléatoire si la chaîne B\$ est plus longue que la chaîne A\$: on obtient zéro pour A\$ = "A" et B\$ = "ABC" (ce qui est normal) mais 10 pour B\$="CBA" !

En TB, quand un paquet de DATA dépasse une vingtaine de Ko, on se retrouve dans une zone mémoire échappant à tout contrôle et provoquant des affichages totalement aberrants... Tout peut redevenir normal quand on dépasse ce "triangle de Borland".

En TB, dans une application gérant l'ensemble des frappes au clavier par INKEY\$, l'appui sur la lettre Q est interprété comme la fonction Quit... et on quitte effectivement le programme pour se retrouver dans le DOS ! L'effet désopilant est garanti quand on vient de travailler pendant plusieurs heures et que tout est perdu.

En TB et en QB, le retour sur une réponse refusée après LINE INPUT REP\$ (très utile pour modifier le texte sans avoir à le retaper in extenso) est, tout simplement, impossible : tout déplacement du curseur provoque l'effacement des caractères.

En TB et surtout en QB, la longueur des lignes logiques n'est pas toujours tout-à-fait de 256 signes, ce qui peut être gênant quand on reprend un programme écrit en GWB bien tassé...

En TB, la gestion de la 25e ligne reste un mystère.

UNE QUESTION, POUR FINIR !

Quelqu'un connaît-il un Basic idéal pour des littéraires ? On dit beaucoup de bien du Basic 1000 D chez Atari (600 F ttc), particulièrement malin en calcul formel. L'est-il aussi pour manipuler les textes et simplifier les affichages ? Est-il licite de rêver de variables chaînes comportant des attributs d'affichage tels que la couleur ou la taille des caractères ou toute autre fantaisie, comme la possibilité de justifier les textes en rédigeant un programme, bien dans l'esprit de ces incorrigibles et insolubles littéraires...

TABLEAU RÉCAPITULATIF

	GW BASIC interprété MICROSOFT	TURBO-BASIC compilé BORLAND	QUICK-BASIC compilé MICROSOFT
Editeur			
prix	inclus	± 870 F.	± 870 F.
taille fichiers principaux	70 Ko	260 Ko	600 Ko
mémoire vive		320 Ko	384 Ko
transfert de GWB vers TB ou QB		facile	facile
essai	direct	lent	rapide
Ctrl-break	par défaut	à activer	par défaut
édition		type Wordstar	couper/coller
compilation		tout programme jusqu'à 64 Ko	limitation par fichier intermédiaire
exécution		fichier .EXE	fichier .EXE ou avec BRUN
copie d'écran	LCOPY	INTERRUPT 5	procédure spéciale
variables dans DRAW ou PLAY documentation	facile	difficile bon manuel F1 = aide	difficile manuel touffu aide en ligne

1/11/90

Jean-Louis MALANDAIN
Chargé d'Etudes au CIEP-Belc
Membre du Bureau national de l'EPI