

GROS PLAN

L O G I C I E L



TP 6 propose désormais l'environnement de développement IDE.

Turbo Pascal 6.0

A l'heure où la programmation orientée objet semble emporter les suffrages, sinon de ceux qui font du développement, du moins de ceux qui ne font qu'en parler, le Pascal, lui aussi, cède à cette mode, offrant toutefois un environnement de développement beaucoup plus agréable.

Depuis le lancement de Turbo Pascal 4.0, chaque année voit la sortie d'une nouvelle version du langage qui a fait la fortune de la société Borland. Fidèle au produit qui est à l'origine de son succès, Borland propose donc aujourd'hui Turbo Pascal 6 (TP 6), et la version développeur qui lui correspond, Turbo Pascal 6 Professionnel.

Turbo Pascal propose maintenant le même environnement de développement intégré (en anglais IDE, pour *Integrated Developer's Environment*) que Turbo C++, qui faisait pâlir d'envie tous les développeurs Pascal ! L'interface en mode texte ne dépaysera pas les utilisateurs de Windows 3.0 : multifenêtrage et gestion de la souris. L'éditeur de texte dispose (enfin !) des fonctions associées au couper/coller et au presse-papiers.

La meilleure surprise de Turbo Pascal 6 est sans doute l'apparition de Turbo Vision (même si l'évolution de l'interface était, à vrai dire, logiquement prévisible). Il s'agit d'une bibliothèque de fonctions qui permet de développer très rapidement le squelette d'une application en la dotant d'une interface semblable à celle de

l'IDE. C'est d'ailleurs avec Turbo Vision qu'a été réalisée l'interface de Turbo Pascal, ce qui constitue un gage de la qualité de cette boîte à outils.

Le programme d'installation est conforme au classicisme efficace des produits Borland, qui décompacte et installe rapidement les cinq disquettes (au format 5"1/4, 360 Ko) du logiciel. Il présente toujours le même petit défaut : si l'installation est faite depuis un disque dur, l'accès au programme de décompactage (*Unzip Exe*) est impossible, et il faut le copier manuellement au préalable.

Une interface multifenêtre

Une première surprise attend les habitués : le programme TPinst a disparu, et la configuration (y compris des couleurs) est dorénavant totalement intégrée à l'IDE. S'il est théoriquement possible de travailler à partir d'une disquette 720 Ko, le disque dur est bien évidemment quasi-indispensable. En revanche, pour utiliser l'IDE, Turbo Pascal 6, s'il le faut, sait se contenter d'une configuration digne de la préhistoire : un compati-

ble XT, avec au moins 512 Ko de mémoire vive.

A tout seigneur tout honneur : c'est principalement la puissance de son interface de développement qui a fait le succès de Turbo Pascal. Elle accède enfin à maturité : adieu, les commandes Wordstar de manipulation de l'éditeur, et que vive la souris ! L'éditeur autorise l'ouverture d'autant de fenêtres programmes que la mémoire le permet, avec toutes les manipulations habituelles : déplacement, modification de taille, affichage en cascade ou par juxtaposition, le tout sur un simple « clic » de souris. Les menus font appel à des boîtes de dialogue qui font bon usage de boutons radio, cases cochées et autres ascenseurs : le tout est bien conçu, souple et rapide.

L'éditeur de texte dispose d'un atout non négligeable que la documentation papier ne met pas vraiment en valeur, puisqu'elle ne le mentionne nulle part ! En effet, l'utilitaire Temc (voir *infra*) permet d'affecter aux touches du clavier des macro-commandes.

L'aide contextuelle de l'IDE est complète, qui concerne non seulement la syntaxe du langage, mais aussi toutes les bibliothèques fournies, y compris Turbo

Vision. De nombreux exemples y sont proposés, qui peuvent être insérés directement dans le presse-papiers.

Une nouvelle option de compilation assure la génération de code spécifique au processeur 80286, sans qu'on puisse constater d'améliorations sensibles en taille comme en vitesse d'exécution.

Néanmoins, quelques ombres doivent figurer à ce tableau. Un regret d'abord : pourquoi n'avoir pas intégré la possibilité de réduire les fenêtres en icônes (comme dans la nouvelle version de Turbo Debugger) ? La fenêtre de sélection de fichiers ne mémorise pas, dans la liste des fichiers précédemment ouverts, le chemin d'accès complet, et le travail sur plusieurs répertoires se trouve donc alourdi. Plus grave enfin : certains raccourcis claviers (Ctrl-Ins, Ctrl-Del) fonctionnent de manière erratique sur le clavier 84 touches du compatible AT utilisé pour nos essais, le problème se produit même sans avoir chargé au démarrage le programme Keybfr, alors que d'autres logiciels utilisent sans difficulté aucune les mêmes raccourcis. Très logiquement, ce problème se retrouve dans les applications utilisant Turbo

LES DEBOGUEURS

Mettre au point un programme réalisé en Turbo Pascal – manière pudique de dire : en corriger les erreurs ! – peut se faire directement dans l'IDE grâce au débogueur intégré, ou avec Turbo Debugger. Jusqu'à Turbo Pascal 5, la mise au point d'un programme consistait à inclure aux endroits *ad hoc* l'affichage du contenu d'une variable, ou tout simplement un message indiquant que telle procédure était exécutée.

Un débogueur répond de manière plus efficace aux mêmes besoins : connaître à un instant donné le contenu d'une variable, ou savoir si le programme passe ou non par telle ligne du code. On n'a donc plus besoin de modifier le programme source. Dans les deux versions, on dispose des points d'arrêt qui stoppent l'exécution

du programme à une ligne donnée, et d'une fenêtre de suivi affichant la valeur de variables ou d'expressions.

Le débogueur externe offre de nombreuses possibilités supplémentaires : la visualisation de toutes les variables globales ou locales d'un programme, de la hiérarchie des objets ou des appels de procédures. La principale qualité du débogueur externe réside dans la possibilité de mettre au point un programme sans que la présence du programme de débogage n'interfère dans l'exécution du programme testé. On peut réaliser ceci en débogant en mode 286 ou 386 si l'on dispose d'un tel processeur ou, mieux encore, en mettant au point un programme tournant sur un PC à partir d'un autre PC, les deux reliés par une liaison série.

Vision. Mais il ne s'agit, au fond, que de points de détail, ne nuisant en rien à l'excellente qualité de l'ensemble.

Les diverses améliorations se payent par une augmentation considérable de la taille du programme qui double (de 150 Ko pour la version 5.5, à quelque 325 Ko). Heureusement, la mémoire disponible pour les programmes reste importante, grâce à une gestion efficace de la mémoire paginée : faute d'avoir ce type de Ram, en revanche, les performances sont un peu moins satisfaisantes, puisqu'il faut patienter le temps que soit lu et écrit un fichier temporaire sur le disque dur. Mais, avec ou sans mémoire paginée, la taille maximale des programmes qu'on peut compiler ou exécuter depuis l'IDE ne change pas, ou peu.

Comme dans les versions précédentes (depuis la 5.0),

la mise au point est facilitée par le débogueur intégré, qui permet maintenant de préciser les points d'arrêt conditionnels (voir encadré ci-dessus), comme de visualiser dans une fenêtre les registres du processeur (nouveau mineure, mais agréable).

Un assembleur intégré

Les changements des spécifications du langage, pour être moins frappantes que celles de l'IDE, n'en sont pas moins substantielles. Cependant, hormis la gestion de la mémoire (plus précisément du « tas » ou *heap*), la compatibilité est assurée avec la version précédente de Turbo Pascal : la compilation d'anciens programmes ne pose absolument aucun problème, de même, d'ailleurs, que la réutilisation de procédures.

Les fanatiques de l'assembleur vont pouvoir librement assouvir leur passion pour le code machine, puisque Turbo Pascal dispose désormais d'un assembleur intégré : plus besoin de procédures externes ni de la fonction *Inline*. Il suffit de la séquence *ASM*, suivie des instructions assembleur, puis de la directive *End* ; nul ne s'étonnera des ressemblances avec le Turbo Assembler de Borland. Une fonction ou une procédure dont l'en-tête contient la directive *Assemble* peut être alors entièrement rédigée, si besoin est, en assembleur.

La clause « *private* » complète la panoplie de la gestion des objets : elle permet de rendre « locaux » certains des champs et méthodes d'un objet à l'intérieur du module le contenant. Par ailleurs, la gestion des flux (*Stream*) d'objets ne nécessite plus l'utilisation de fichiers binaires externes – qui donnaient un aspect « bricolé » aux flux de la version 5.5.

Le seul changement appelant une modification des programmes écrits pour les versions antérieures concerne la gestion du « tas », c'est-à-dire la portion de la mémoire vive à laquelle un programme en Pascal accède en créant dynamiquement des pointeurs. La structure de la liste des emplacements vides dans le tas a été modifiée afin d'en réduire la fragmentation.

L'évolution de la syntaxe du Turbo Pascal au fil des versions conserve une bonne compatibilité, tout en offrant aux développeurs la possibilité de disposer des dernières améliorations en techniques de programmation. Avec sa version 6, Turbo Pascal permet d'accéder au sein d'un même environnement, grâce à une syntaxe homogène, aussi bien à la programmation objet qu'à l'assembleur.

La véritable révélation de

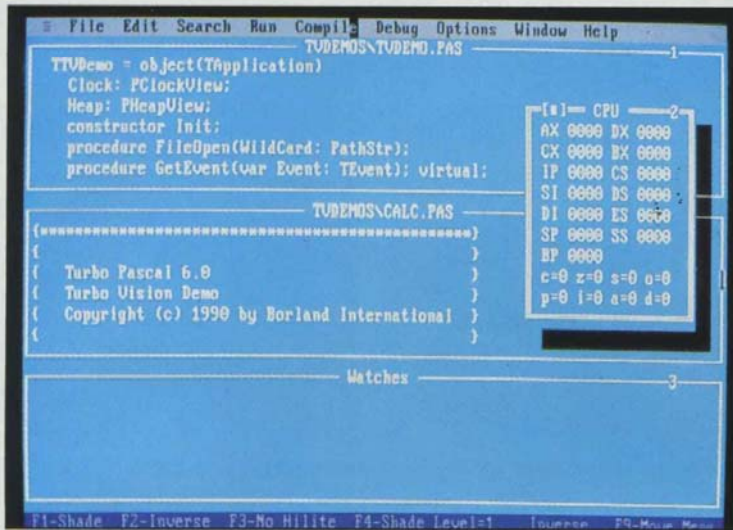
Turbo Pascal 6 reste incontestablement la boîte à outils (Toolbox) Turbo Vision. Avec elle, Borland porte là un coup redoutable aux sociétés diffusant ce type de produit.

Apparente simplicité...

Les concepteurs de Turbo Vision ont fait d'entrée un choix crucial : ils n'ont pas tenté de réaliser une boîte à outils universelle avec trente six mille procédures faisant tout... et le reste ! Turbo Vision sert à créer des squelettes d'applications, en fournissant les objets et les méthodes pour créer et manipuler un « espace de travail » semblable à celui de l'IDE. La création de barres des menus, de fenêtres avec barre de défilement, de boîtes de dialogue, est d'une simplicité désarmante. Quant à la gestion de la souris, elle est totalement transparente : dès qu'une fenêtre est créée, un simple « clic » suffit pour la fermer, la déplacer ou la redimensionner. Quelques modifications apportées aux fichiers donnés en exemple permettent de construire en un temps record l'arborescence des menus d'une application, y compris de ses boîtes de dialogue. La comparaison avec Windows est immédiate mais, pour l'instant, il n'est pas possible de créer des applications Windows avec TP 6 (sans doute pour la prochaine version 7).

...mais complexité réelle

Au premier abord, le résultat semble miraculeux : le développement d'applications d'aspect « professionnel » en quelques lignes de codes. Les développeurs, pourtant, n'ont pas à craindre le chômage tout de suite, car il y a un saut considéra-



Véritable innovation : la boîte à outils Turbo Vision.

ble pour passer du squelette de l'application à l'application elle-même (comme on l'oublie trop souvent). L'habillage de l'application nécessitera une connaissance détaillée de la logique adoptée par Turbo Vision, une bonne maîtrise de la programmation objet et, enfin, d'exceller dans la manipulation des pointeurs. La programmation avec Turbo Vision n'est pas sans rappeler la programmation sous Windows... et ses difficultés.

Turbo Vision manipule des « vues », c'est-à-dire des objets obtenus par instanciation ou dérivation des objets fournis par la librairie, et correspondant à des éléments affichés à l'écran. Au départ, il faut créer un objet générique de type application, à l'intérieur duquel on insère différentes vues. Il existe des objets prédéfinis correspondant à l'espace de travail, aux menus, à la barre de statut, boîtes de dialogue et fenêtres. Les véritables problèmes commencent au moment de relier ces différentes vues, ce qui revient, avec Turbo Vision, à les faire communiquer.

Il faut abandonner tout souvenir de la programmation procédurale « classique », où des procédures s'enchaînent les unes aux au-

tres. L'utilisateur d'une application réalisée avec Turbo Vision intervient dans le déroulement de l'application en créant des événements (frappe de touche, déplacement de la souris). Cet événement est transmis par l'application, de vue en vue, jusqu'à ce qu'il arrive à la vue concernée. Chaque type de vue dispose d'une procédure chargée de déterminer quelle action entreprendre suivant le type d'événement qui lui est transmis. Développer une application avec Turbo Vision revient donc largement à créer ces procédures de gestion des événements. Cependant, pour le programmeur, il n'est pas toujours aisé de saisir le cheminement d'un événement au sein d'une hiérarchie complexe d'objets.

On présente souvent les boîtes comme une panacée : c'est oublier que, même pour les mieux faites d'entre elles, il faut consacrer un temps considérable à en saisir les modalités d'utilisation, et à assimiler la syntaxe des procédures. C'est à l'usage que se révèle la souplesse de Turbo Vision. A titre d'exemple, il nous a suffi de moins d'une journée pour réaliser un « mini IDE », permettant d'éditer, de compiler (sur disque) et d'exé-

ter un programme Pascal. Pour ce faire, nous avons simplement repris le programme exemple d'éditeur de texte, pour y ajouter les options de menu permettant la compilation et l'exécution du contenu d'une fenêtre, sans la moindre difficulté. Seule la réalisation des procédures appelées par ces options a posé quelques problèmes mineurs à vrai dire. Sans Turbo Vision, et pour un résultat bien moins élégant (pas de gestion de la souris, pas de multifenêtrage), une dizaine de jours nous aurait été nécessaire.

Des utilitaires puissants

Différents utilitaires sont fournis avec Turbo Pascal dans sa version standard. Le compilateur « ligne » permet de compiler, directement depuis l'indicatif Dos, un programme Pascal contenu dans un fichier Ascii : caractéristique utile, notamment lorsque l'espace mémoire exigé par le programme interdit sa compilation à partir de l'IDE.

Le compilateur ligne est aussi utile pour ceux qui voudraient utiliser leur propre éditeur de texte, d'autant que l'aide contextuelle est utilisable hors de l'IDE : un programme résidant (Thelp) permet de visualiser les fichiers d'aide tant de Turbo Pascal que de Turbo Assembler, Debugger ou Profiler.

La simplicité « spartiate » de l'éditeur intégré dans l'IDE des précédentes versions justifiait le recours à un éditeur externe. Son amélioration rend ce recours inutile, à notre sens, d'autant plus que l'utilitaire Temc (*Turbo Editor Macro Compiler*) permet de modifier l'assignation des touches, et surtout la création de macros enchaînant plusieurs commandes de l'éditeur. Temc compile un fichier Ascii contenant le texte des ma-

cro, rédigé dans une syntaxe qui rappelle celle du Pascal, et ajoute le code de ces définitions dans un fichier de configuration de l'IDE. Temc ne connaît pas les commandes interactives, mais il permet l'insertion d'un texte fixe. L'exemple donné dans l'encadré ci-contre associe à la commande Ctrl-A-P l'insertion d'une en-tête de procédure. Sans concurrencer un éditeur Ascii spécialisé comme Brief, avec Temc, les programmeurs pourront personnaliser l'IDE, ce qui constitue un atout supplémentaire.

Un reproche cependant, il est anormal que les utilitaires (hormis le compilateur ligne) ne soient pas décrits autrement que dans des fichiers. L'impression des fichiers .DOC occupe plus de 40 pages : c'est pour le moins excessif et même gênant quand, de surcroît, le fichier en question est imprécis - comme l'est celui qui traite de Temc !

Une documentation complète

La documentation se compose de quatre volumes qui totalisent, avec les index, plus de mille pages. Le manuel de l'utilisateur est une introduction à toutes les facettes de Turbo Pascal, et de son environnement ; le guide du programmeur donne la syntaxe du langage Pascal, et des informations plus techniques sur l'organisation interne des données ; un autre volume contient un dictionnaire de toutes les fonctions et procédures du langage ; un dernier volume, enfin, fort volumineux, est entièrement consacré à Turbo Vision.

Fort complète, riche en exemples, la documentation est bien faite, comparée à bien des manuels (y compris ceux de logiciels destinés à un plus large public). Néanmoins, en dépit de l'évolution considérable du produit

depuis sa version 5, il n'y a pas eu de refonte dans la présentation du langage, mais simplement des ajouts de chapitres et de paragraphes aux endroits adéquats. Cela peut nuire, nous semble-t-il, à une bonne vision d'ensemble pour un utilisateur débutant. Le meilleur exemple concerne la programmation objet. Le chapitre qui lui est consacré est un modèle de clarté et d'intelligence, mais il est littéralement « collé » à la suite des chapitres qui traitent de la programmation classique, sans que soit donnée dès le départ une vision globale. Enfin, on peut espérer que les nombreuses erreurs qui entachent le chapitre d'initiation à Turbo Vision seront corrigées dans la traduction française.

Une version « professionnelle »

La version professionnelle du logiciel offre trois programmes supplémentaires et différents utilitaires. Cette qualification de « professionnelle » est appropriée, car ce package ne se justifie que pour un développeur : comme dans les versions antérieures, il offre un débogueur indépendant (Turbo Debugger), et l'assembleur Borland (Turbo Assembler).

Turbo Profiler, une nouveauté, permet d'évaluer les performances d'un programme et de déterminer

quelles sont les parties à optimiser. Inutile toutefois de compter sur un miracle : le « profileur » n'optimise pas les programmes tout seul ! Il se contente d'afficher quelle proportion du temps d'exécution d'un programme prend telle ligne ou telle procédure.

Pour pouvoir l'utiliser, il suffit d'avoir compilé sur disque un programme Pascal en y incluant les informations de débogage, exactement comme pour mettre au point un programme avec Turbo Debugger. Le profileur ne sert, en fait, qu'à évaluer les performances des portions de code non interactives, car mesurer le temps mis par des procédures attendant des entrées de l'utilisateur serait bien évidemment dépourvu de sens. Pour accélérer la vitesse d'un programme, il suffit alors de modifier la partie désignée comme « lente » par l'utilitaire efficace, mais d'un usage très spécifique. Ces trois programmes sont accompagnés d'une documentation complète, particulièrement Turbo Assembler, qui est livré avec un manuel d'initiation à l'assembleur.

Par rapport à la version professionnelle de Turbo C++, qui contient les mêmes programmes, on dispose en plus de Turbo Drive, une version du compilateur ligne gérant la mémoire étendue. Le programme compilé ainsi ne reconnaît pas la mémoire

```
UNE MACRO TEMC

SCRIPT TPIDE;
/* Macros clavier Turbo IDE */
/* TPIDE.MAC */
/* Le 8/1/91 */
Macro Procéd
/* insère le texte d'un entêt de procédure */
  InsertText("PROCEDURE;\nVAR;\nBEGIN\n\nEND;\n");
  CursorUp; CursorUp; CursorUP; CursorUp; CursorUP;
  RightOfLine;
  CursorLeft;
END;
/* Assignation des Touches */
Ctrl-A+^P:Proced;
```

étendue (seule la gestion de la mémoire paginée est déjà prévue dans Turbo Pascal ; c'est au programmeur d'écrire son propre gestionnaire de mémoire étendue), et le compilateur utilise la mémoire étendue pour traiter des programmes très volumineux.

Des apports indéniables

On est parfois saisi d'un doute légitime face à la prolifération de ces nouvelles versions, plus encore peut-être quand il s'agit d'un langage de programmation : la perte de temps que constitue l'acquisition des connaissances nécessaires à la bonne maîtrise du logiciel et des modifications du langage sera-t-elle compensée par un réel gain en puissance et en pro-

ductivité ? Les apports de cette version 6.0 sont indéniables. L'IDE, à lui seul, autorise un gain de temps considérable.

L'éventail des possibilités offertes par le langage est très vaste ; et la boîte à outils Turbo Vision, une fois maîtrisée, constitue un outil d'une qualité indéniable, qui met à la portée de tout programmeur un Windows-like... en mode texte.

Logiquement, une version de Turbo Pascal, compilant des applications Windows, est annoncée pour cette année. Si Borland réussit le portage de son compilateur dans cet environnement sans trop diminuer sa rapidité, le paysage du développement sous Windows devrait sérieusement changer.

J. Chodkiewicz

Pour plus d'informations, cerclez 41 & service lecteur télématique

SOFT & MICRO

Une somme inestimable d'informations et de références

ABONNEZ-VOUS

au mensuel leader de la micro-informatique professionnelle

Reportez-vous à l'encart abonnement situé page 35