

# G-Pascal puts power in home

**Like many home computers, the Commodore 64 has been restricted by software limitations. Ian Webster discusses a new implementation, G-Pascal, which unleashes its power.**

ONE of the enduring problems of home computers has been the software environment supported by the hardware. As the cost of hardware has decreased and designers have been able to improve the quality of the hardware environment with better graphics, color and sound, home computers have become much more attractive to consumers.

Unfortunately, the software environments supplied with the machines have not developed as well, particularly when manufacturers decide new models must be compatible with previous machines. Control of the new features usually requires a blizzard of PEEKs and POKEs, a clear understanding of the internal memory map of the machine and familiarity with machine code.

Often when extensions to the Basic interpreter are provided, the space constraints in ROM make the code size of the extensions rather than the speed of the machine the most important consideration. Programmers quickly become frustrated with the Basic interpreter, particularly if they are writing games which require reasonable execution speed to be playable.

Two Melbourne programmers, Nick Gammon and Sue Gobbet, recognised the problem on the Apple microcomputer and wrote a system for programmers who wanted to write games. The system was based on a p-code Pascal implementation modelled on the UCSD Pascal programming environment.

The Pascal implementation supporting all the features of the Apple environment offered by the Commodore 64 is one of the worst available, considering the power-rich features the hardware supports.

Graphics and sound are almost beyond the reach of novice programmers using the Commodore 64 and a playable game would be difficult without a lot of experience. Gammon and Gobbet have just released an implementation of their G-Pascal game programming system for the Commodore 64 that provides outstanding support for all of the features of the system.

```
CONST COLOUR = 1; POINT = 2; YELLOW = 7;
BEGIN
  DEFINESPRITE (128, $FFFFFF, $F0000F, $F0000F, S $FFFFFF);
  SPRITE (1, POINT, 128, 1, COLOUR, YELLOW);
  MOVESPRITE (1, 50, 50, 256, 256, 180);
END.
```

*A six-line program to define and move a sprite*

G-Pascal supports a memory-resident development system with a powerful text editor to create the source code and a high-speed compiler which implements a substantial subset of Pascal. Seventy-six extensions for control of the Commodore 64's features are available. The compiler supports independent, relocatable modules and a runtime package is available for people who want to develop standalone applications or games.

Support for graphics, sprites and sound is superb and the attention to

Spritecollide, Groundcollide and Spritefreeze.

All these commands have a parameter list. The sound commands are as comprehensive and provide complete control of the SID chip.

Advanced game programmers require control of the interrupt routine and the position of the raster line of the screen at any time to avoid flickering as the image is drawn on the screen. G-Pascal provides this control to the programmer as well as the capability to use assembly language subroutines.

The Commodore environment is idiosyncratic because of its use of intelligent peripherals. G-Pascal has overcome the problems that seem to plague Commodore system software developers and supports all the standard file commands.

The text editor is line-based and offers enough feature to satisfy most users. A Debug and Trace mode is available to display the information needed to track down difficult program bugs.

The manual is a reference document and does not teach the buyer to program in Pascal, but describes the operation of the G-Pascal system and all the commands. G-Pascal is a thorough, considered implementation of a game programming environment for the Commodore 64.

It unleashes the power of the machine and provides a superior program development system that will enhance the quality of any user's programming with a Commodore 64.

You may have heard that Australian programmers are as good as any. Here is the evidence.

G-Pascal compiler Version  
3.0 Ser= 4321

Written by Nick Gammon and Sue Gobbet, Copyright 1983 Gambit Games, P.O. Box 124 Ivanhoe 3079, Victoria, Australia.

P-codes ended at 412E  
Symbol table ended at C026.

(C)ompile finished: no Errors.

(E)dit, (C)ompile, (D)ebug, (F)iles,  
(R)un, (S)yntax, (T)race, (Q)uit?

*The G-Pascal environment*

control of the machine environment has created a system that should become very popular with programmers trying to use the features of the Commodore 64. All graphics modes are supported, including high resolution and control of the different memory banks available for video memory.

The sprite commands are possibly the most comprehensive yet written for any machine on the market. They include Definesprite, Positionsprite, Movesprite, Animatesprite, Spritestatus,