TURTLE GRAPHICS

History

A graphics package, called Turtle Graphics, has been part of the UCSD p-System from its earliest stages. The implementation for the TI 99/4A, however, lacks this package. This is of course due to the fact that this p-system was developed for the 99/4, which didn't have any bit-map mode.

But the 4A has, and hence the hardware doesn't prevent graphics. But one problem remained. The 99/4A uses VDP RAM as an alternate code pool, not giving enough space for the extensive memory requirements imposed by the bit-map mode. Unfortunately the TI system was delivered without the kernel unit, and therefore prevents easy access to system resources.

Now this problem has been circumvented. After figuring out where the system has some pointers used for memory management, conversion to bit-map mode turned out to be possible, without endangering the p-system execution environment.

Memory reservation

When an application that uses turtle graphics is loaded, the graphics package attempts to reserve the required VDP RAM space before the main program starts. But if the application is large, this might be impossible, since the required memory might already be occupied by program code.

When this happens, the user is notified, and the program halts.

The solution is to reserve the memory prior to loading the application. A special utility, called BITRESERVE, is provided to perform this. BITRESERVE allows allocation and deallocation of bit-map memory, as well as showing the amount of VDP RAM available.

Another solution, which doesn't require any special action every time your program is to be loaded, is to change the code type of your program. If the code type of your program is M_9900, rather than M_PSEUDO, that program will be loaded in the alternate code pool (32 K RAM bank) from the beginning. Thus it will not intrude on the VDP RAM required for bit map mode.

Introduction

The graphics package is reached by including the statement 'uses turtlegraphics' in your program. The routines then provided are more or less a copy of the capabilities offered by the same unit for Apple Pascal II.1. Some parts are influenced by the unit usually accompanying the UCSD p-system version IV. Finally, some changes has been imposed by the somewhat restricted color resolution provided by the TMS 9918A/9929.

General

The screen is considered a rectangle, 256 pixels wide and 192 pixels high. Each pixel is reached with its coordinate, with (0,0) being in the lower left and (255,191) in the upper right corner. By default, the background color is transparent, which allows the standard p-system backdrop color (cyan) to show through. The foreground color is black.

Grafmode

The procedure grafmode enables the graphics mode. The defaults are loaded, and the screen is erased.

Textmode

By calling textmode, the system returns to the default screen, giving 24*40 characters at a time. The text shown on the screen when grafmode was called is restored, although the definitions of the characters 128..255 are erased. NOTE! Calling textmode twice, without any grafmode in between, leaves the system in an indeterminate state. The same is valid for the opposite combination.

Initturtle

When already in graphics mode, i.e. after calling grafmode once, initturtle clears the screen and sets the defaults, without returning to text mode. Initturtle may be called any number of times.

Viewport

All screen activities are constricted to the viewport concept. The viewport is the active part of the screen. By calling viewport, you may limit the actions caused by drawing lines etc. to the active rectangle. The proper argument sequence is viewport(left,right,bottom,top). The

viewport defaults to the entire screen
(0,255,0,191).

Pen color

Calling pen_color sets the desired color. The type screen_color is accessible to the application. It contains the colors allowed. They are: transparent, black, green2, green3, blue1, blue3, red1, cyan, red2, red3, yellow1, yellow3, green1, magenta, gray and white. Where numbers are used to designate different shades of the same color, one (1) is used for the darkest shade etc.

Pen mode

Although the color used is set by pen_color, pen_mode determines if drawing is done at all. There are nine modes, but some gives the same effect.
INVISIBLE: No drawing occurs.
SUBSTITUTE: Drawing occurs. The foreground color is changed to the active color.
OVERWRITE: Same as substitute. Included only for compatibility reasons.
UNDERWRITE: Drawing occurs. The color is changed only if the pixel wasn't occupied previously.
COMPLEMENT: If the pixel isn't set already, set it. Then use the background color, complement it, and reinsert it as the new foreground color.
If the pixel is set already, just complement the foreground color.
NONE: Same as invisible.
DRAW: Drawing occurs. No change of color.
ERASE: Erases eventually set pixels. Doesn't change the color.
REVERSE: Inverts the pixels. Doesn't

change the color.

Since the modes none, draw, erase and reverse doesn't affect the colors, they are faster than their colored counterparts.

Move

Moves the turtle the specified distance in the current direction. Leaves a trail, being the straightest possible line, on the screen, if pen_mode is set to something that enables drawing, and the turtle moves in the current viewport. If the turtle starts and/or stops outside the viewport, but passes it on the way, only the part of the trail that is inside the viewport is drawn. The default position of the turtle is in the middle of the screen (128,96), heading to the right (0 degrees).

Moveto

Same as move, but moves to a specified coordinate.

Turn

Turns the turtle the specified angle. Only integer angles are used, one revolution comprising 360 degrees. Positive angles results in turns to the left or counter-clockwise.

Turnto

Same as turn, but turns directly to the specified angle. 0 is to the right, 90 upwards, 180 to the left, 270 downwards and so on. Negative angles are also allowed, with -90 degrees giving the same result as 270 degrees, etc.

Turtle x, turtle y and turtle ang

By calling the functions turtle_x, turtle_y and turtle_ang, the current position and heading of the turtle can be determined.

W_char

Writes one character on the screen. The character is placed with its lower left corner at the current turtle position. The position is then incremented by six, which is the character width in text mode. The character definitions are the same as used before calling grafmode. NOTE! If you want to change the appearence of some character by using set_pattern in the support unit, this must be done prior to calling grafmode. Calling any routine in the support unit, except set_chr_color and joy, gives incorrect results and/or leaves the system in an indeterminate state.
NOTE! Only character codes 0..127 are available when the graphics mode is active.
Attempts to place a character outside the current viewport gives no visible result, but the current position is still incremented. This is also true even if it gets outside the physical screen.
W_char doesn't adhere to the current pen_color or pen_mode, at least not yet. Writing always takes place, using the current foreground and background colors.

W_string

Same as w_char, but writes an entire string. No scrolling is done, in any direction.

Hardcopy

Copies the screen, without color information, to the printer. The output is adapted to single density resolution for Epson-compatible printers.

Additional information

When colors are complemented, the following scheme is used.

Transparent	-> Gray
Black	-> White
M Green	-> M Red
L Green	-> D Red
D Blue	-> L Yellow
L Blue	-> D Yellow
D Red	-> L Green
Cyan	-> Magenta
M Red	-> M Green
L Red	-> D Green
D Yellow	-> L Blue
L Yellow	-> D Blue
D Green	-> L Red
Magenta	-> Cyan
Gray	-> Transparent
White	-> Black

Since the graphics mode introduces a turmoil in VDP RAM, all screen related normal procedures will fail to behave properly. Keyboard input with read is impossible, as well as screen output with write. In the former case, use bufscan in the extrascreen unit. In the latter, use w_string.

This movement of tables also disables the proper function of the screen left and screen right function keys. NOTE! These keys are not made inactive, but their use will usually corrupt the color table used for graphics.