Animating OPL

The final installment of the graphics programming tutorial, by Damian Walker.

OPL is a very user-friendly programming language. One way in which it helps you is by cleaning up after you, when you've forgotten to do so. Earlier in the tutorial I explained the gCLOSE statement, and why you should use it to free up the space taken by a bitmap when you no longer need it. The same is true of sprites and the bitmaps that make them up.

While failing to do this in our own program has left no ill effects, let's imagine that this bouncing ball demonstration was to be part of a larger program, say, as a title screen for a game. What would the consequences be of not clearing things up? Well, for a start, we've create a new sprite each time we run the program, but at the end we're leaving it on the screen. If the Bouncer procedure were called repeatedly, we'd have a growing number of bouncing balls on the screen. In addition to that, our loading its bitmaps over and over again would take up more and more memory.

For this reason I recommend a clean up operation before the program finishes. The following procedure does the trick:

```
PROC CleanUp:
     SPRITEDELETE:(sprite&)
     BITMAPUNLOAD:(ball&(1))
     BITMAPUNLOAD:(mask&(1))
     BITMAPUNLOAD:(ball&(2))
     BITMAPUNLOAD:(mask&(2))
     BITMAPUNLOAD:(ball&(3))
     BITMAPUNLOAD:(mask&(3))
ENDP
```

SPRITEDELETE not only wipes the sprite from the display, but also from memory altogether. To bring it back would involve not just calling SPRITEDRAW, but SPRITECREATE and SPRITEAPPEND as we did in our LoadBall procedure. It is an inconvenience that there is no SPRITEHIDE procedure to complement the SPRITEDRAW, for those situations when you want to temporarily remove a sprite from the screen. The BITMAPUNLOAD procedure are similar in effect to the gCLOSE statement we used to close the floor tile bitmap.

Further Development

Obviously sprites are ideal for arcade games. The only thing you need to learn before progressing to writing games is how to read the keyboard and respond to it. An easy way of doing this is to use the KEY function, which scans the keyboard, but doesn't stop and wait for a key like GET does. It simply returns 0 if no key is pressed. There are more advanced techniques, particularly if you want to use the pen or to respond to EPOC's system events, but these are the subject of a future tutorial.

Much more can be done with the graphics themselves, too. You're not limited to Sketch. There are free images available on the Internet that you could adapt to your needs, and if you're competent with a desktop graphics package like Photoshop you can probably create much more impressive bitmaps for your programs. One thing to remember is that for greyscale images a relatively small palette can suffice, compared to colour. The four greyscales usually used on the monochrome EPOC machines can achieve good results, particularly if you can use dithering for subtle shading effects.

There leaves the question, what sort of game can one create with these techniques? I wouldn't want to limit your imagination, but you would need to consider a few things before getting too ambitious. First of all, OPL is an interpreted language, and though it is a fast one, there is still a limit to how much processing you can do before things start to slow down. And even without the processing time taken by your own program, having a lot of sprites simultaneously animated on the screen can take its toll on performance.

But there are still plenty of things you can do with a handful of sprites on a static background. I hope that you have found this tutorial useful, and that it has provided food for thought. Please do contact me and let me know, especially if your new-found knowledge leads to any interesting new projects you'd like to show off!

EPOC ENTERTAINER

Issue 6: May 2008

Edited by Damian Walker

This month...
No Mans Land

Welcome to the May edition of EPOC Entertainer. This month the magazine includes a review of the excellent No Mans Land, by Great Ape Software: our first five-star review! There will, of course, be more reviews in the coming months, and I hope to balance the good games with those that are not so good, to provide a bit of variety.

Also included is the penultimate part of the Connect Four series of reviews, comparing the five games of this type available for EPOC32. I'd be interested to know whether you want to see more head-to-head review series of this type, and what your suggestions would be for future head-to-head subject matter.

Finally, I round off the Animating OPL programming tutorial series. I've had a lot of positive feedback about this series, so it seems there are many people out there who are interested in programming. You'll be pleased to hear, then, that I intend to continue with programming articles in future issues of EPOC Entertainer. Next month will be a short workshop article to discuss issues raised about Animating OPL. The month after that, a whole new programming series will begin!

If you have comments, suggestions, or contributions to make in the form of articles, then please don't hesitate to get in touch at the usual address:

entertainer@snigfarp.karoo.co.uk

The Fount of All Knowledge

Damian Walker introduces the EPOC32 Games Database, from which much of the information for EPOC Entertainer has been taken.

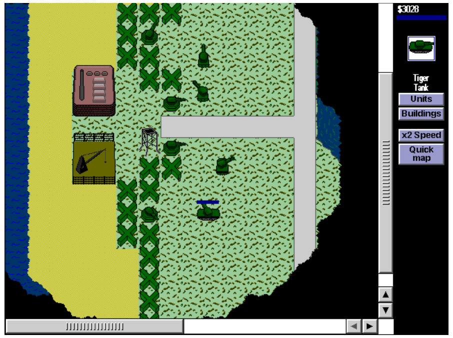
For an enterprise like *EPOC Entertainer*, you need a reasonable breadth and depth of knowledge of the EPOC32 games scene. Back in 2007 I didn't have this, so in the last months of the year I started collecting information on all the games I could find, from sources like the 3-lib CD, old Palmtop magazines, sites like Tucows, and other places on the internet. The result is the EPOC32 Games Database, currently standing at 322 games.

This is available on-line, at the address below. Twelve games are selected at random each day for detailed treatment, including downloads where appropriate. The rest are shown in a summary list. The summary list contains little information, but serves as a good starting point for a web search. Please do make use of this facility: that's what it's there for!

Site	EPOC32 Games Database
URL	psion.snigfarp.karoo.net/psion/gamebase.html

Getting Tanked Up

A review by Damian Walker of No Man's Land, a real time strategy game for EPOC32.



One of the more successful genres on desktop machines has been the Real Time Strategy game. Most readers will have heard of Command & Conquer, Age of Empires and possibly Red Alert. The Psion has few games of this type, but one of those few is No Mans Land, by Great Ape Software.

In No Mans Land you direct a small army of tanks in pursuit of a similarly equipped enemy. The battle ground is populated with oil fields, from which you can obtain resources. With resources you can build more tanks, or instead more buildings like turrets, observations posts and more oil fields.

For sheer game play, No Mans Land has to have top marks. It is very difficult to put the game down once picked up, and I have found myself postponing other things so I can have just a bit more time with it. Except in the first scenario, which is deliberately easy, it can be quite difficult to overcome the computer's defences or resist its attacks. The game also has a virtually unlimited life span: even if you do manage to get through the 22 scenarios, the built-in editor allows you to create new challenges for yourself and for other fans of the game.

The presentation of the game is very good too. While the graphics do not approach the standard of similar games on desktop machines, they are well-drawn and clear on both colour and grey scale screens. There are some good colour images on the title screen, and even a short video clip, for the in-game graphics a simpler approach has been taken. Sound lets the game down a little; the only sound I have heard is a gun shot. I am impressed, though, that the sound gets quieter the further the action is from the part of the battlefield you're currently viewing. The user interface in general is very well implemented. Options are easy to find, and everything seems to be where you expect it.

The game has a very impressive range of features and options. My only criticism is the very limited range of unit types in the game—just two types of tank—but the other features make up for this. As well as the scenario editor mentioned earlier, there many options for setting up custom games. The range of features within the game is extensive too, with options such as automatic production of tanks and repair of buildings.

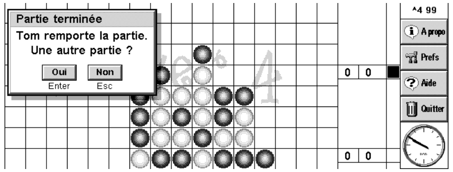
The game manages to impress me further by running on all EPOC32 machines from the Osaris to the Netbook. The speed is quite acceptable on the Osaris and the Series 5 Classic, too. The game takes up quite a bit of memory, around 830K of disk space, and 2.2 megabytes more when running. This is reasonable for a game of this complexity and quality, though.

If you're a fan of Real Time Strategy games you should enjoy No Mans Land, as long as you don't expect the complexity of some of the games available on the desktop. The game is very absorbing, very well written, and now completely free!

By	Great Ape Software
URL	www.greatape.co.uk
Licence	Shareware
Compatibility	Osaris, Revo, Geofox, S5/5mx, S7/netBook
Rating	☆☆☆☆☆

The Right Connections

Damian Walker reviews Power4, in the penultimate article of this review series.


Partie terminée

Tom remporte la partie.
Une autre partie ?

This freeware program from Guillaume Louel is possibly the most interesting game of the bunch. Actually a development of Puissance4, the program's most obvious difference from its rivals (including Puissance4) is the use of a very large board, of 16 columns by 8 rows. While a slight modification to the standard 7×6 board size would have little advantage to compensate for its non-standardisation, such a large difference as this game offers can change the nature of the game significantly. No longer is play confined to a small area, but now when opportunities for forming rows are blocked by your opponent, you have much more space to build alternative plans.

Before going further I must mention that the game is only available in French. This is not a great problem, even for a non-French speaker like me. I was perfectly able to find my way around the program. Personally I find the opportunity to learn a few words of another language an advantage rather than a disadvantage (though there is a limit to this principle—I wouldn't install a game in French or German if English is available).

A major feature not offered by any of the other programs is the ability to select alternative rules for the game. As well as the standard rules, where lines of four can be made horizontally, vertically or diagonally, one can specify at the start of the game that rows can be made only horizontally and vertically, or even only diagonally. While it is questionable whether this would add any enjoyment to the standard game, these restrictions on the formation of rows might work very well on the larger board.

It is possible to specify that 2, 3 or any other number of rows must be formed before a player wins the game. This variation, like those already mentioned, seems very much suited to the large board.

The graphics, designed for the Series 5, are functional but attractive. One frill that Power4 shares with Four In A Line is the ability to change the style of the pieces to suit your taste. The board has “Power 4” in stylised text as a background, but apart from this, the graphics are very spartan, the board being divided by simple black lines. As with Four In A Line, sound is completely absent.

If Power4 had some of the useful features found in Four In A Line, and if it gave the opportunity to play a more standard game, I'd consider it the best program of the five without a doubt. But even without those features, Power4 has to score highly on its versatile rule options alone.

Next month we see the final article in this series, in which Puissance4 is the subject of our review. Also in that article is a comparison table of all the games discussed in this series.