Animating OPL Workshop

An addendum to the recent programming tutorial, by Damian Walker

Over the course of the *Animating OPL* tutorial I've received quite a bit of feedback. One question was raised about the speed of drawing the background, which is quite slow. The reason for this is that I forgot about the gPATT statement when writing the tutorial! You can use it to speed up the drawing of the background immensely, by replacing the DrawGrid procedure with this simpler and more efficient alternative:

```
PROC DrawGrid:
    LOCAL grid%
    grid%=
    ⬀gLOADBIT("\Bouncer\Grid.mbm")
    gUSE 1
    gPATT grid%,gWIDTH,gHEIGHT,3
    gCLOSE grid%
ENDP
```

The gPATT statement simply fills an area of the current drawable with repetitions of the drawable *grid%*. The area is defined in size by gHEIGHT and gWIDTH, with the current drawing position at its top left. To fill a smaller area, you would first use gAT, then reduce the size.

There are further speed improvements that could be made, by creating a temporary drawable, drawing the grid on that, and copying the whole thing to the screen. This works because individual draws to the screen are slower than to a bitmap. But I'll leave it to you to experiment with such techniques.

Another criticism of the *Bouncer* program is that it's stuck in the *\Bouncer* folder. While it can be moved to any drive, if you prefer to develop within another folder, or if you want to develop a full-blown application in *\System\Apps\Bouncer*, then all the file names would need changing. The way around this is to use the *PARSE$* and *CMD$* facilitiy of OPL. The following lines should be inserted in the *Bouncer* procedure, above the call to *DrawGrid:*

```
GLOBAL path$(255),off%(6)
path$=PARSE$(CMD$(1),
                 ⬀CMD$(1),off%())
path$=LEFT$(path$,off%(4)-1)
```

The *CMD$* function returns information about the path of the current program. The *PARSE$* function uses it to find the path of the program and its files, which are stored in *path$*. Now you can change all the references to the *Bouncer* folder to use *path$* instead, as in the example taken from the *DrawGrid* procedure:

```
grid%=gLOADBIT(path$+
                       ⬀"Grid.mbm")
```

There are lines to change in the *LoadBall* procedure as well. Readers who are familiar with the *SETPATH* statement, which in conjunction with *CMD$* would allow us to load files without specifying a path at all, will be wondering why I haven't used that instead. The reason is simple: *BITMAPLOAD&* doesn't honour it, and still tries to load files from the root folder if no path is specified.

That rounds off all the issues I am aware of with regard to the tutorial. If you have any further queries about it, they'll be welcome at the usual email address. Next month will see the start of a new programming tutorial, where Bouncer is amended to respond to keyboard input.

EPOC
ENTERTAINER
Issue 7: June 2008
Edited by Damian Walker
This month...
Adventure Games

I hope the keenest readers accept my apologies for the lateness of this issue: work and illness have both pushed me past the usual deadline, but *EPOC Entertainer* isn't about to fall by the wayside yet!

Last month saw the end of the first programming tutorial, *Animating OPL*, which has been a feature of *EPOC Entertainer* since the very beginning. This has been very popular, and I intend to make games programming a permanent feature of the magazine. One of the strengths of the EPOC platform, which still gives it an advantage over modern PDA machines, is the presence of a built-in, powerful, but easy to use programming language on most models. OPL is easily capable of producing professional quality adventure, RPG, strategy and board games, and with a bit of care, even fast action games are possible. I hope in future issues to bring the programming tutorials to a level where good quality games can be written, and eventually a programming competition might result if there is enough interest. Before moving on to a new tutorial, I've included some corrections and improvements to *Animating OPL.*

This month sees the last of the head-to-head series on Connect 4 games, *The Right Connections*. I'm not sure whether there'll be any more such head-to-head comparisons. What do you, the readers think? Are there any more collections of similar games that you'd like to see given this treatment? Or do you think it gets monotonous to have the same type of game reviewed month after month? Would you prefer to see more one-off reviews? I'd be interested to hear from you on this, so please do get in touch at the email address below.

Another one-off article appears this month, about adventure games: specifically, adventure games using the *FrotzS5* interpreter, of which there are more than 400. This alone more than equals the number of native EPOC games I know about. While I'm not keen to do features on particular non-EPOC or multi-platform games, you can expect to see general articles about running such games in this and future versions of *EPOC Entertainer*. Features on Java games, or on obtaining and running games under emulators such as *Z80*, *E32Frodo*, *EMAME* and *EgnuBoy* may well make an appearance. But don't expect to see detailed information about the games themselves; this is more a subject for magazines and web sites about those emulated machines.

I hope you enjoy this issue of *EPOC Entertainer*. I'm always interested to hear about what you think about each issue. If you have any comments, requests, or offers of help, then please do get in touch at the usual address!

entertainer@snigfarp.karoo.co.uk

In Search of Adventure

A look at text adventure games on the EPOC platform, by Damian Walker

In the late 1970s and early 1980s, computer graphics were very basic, and some computers had no graphics at all. One type of game thrived in this environment: the text adventure. These games would tell a story in which the player was involved; after each paragraph it was the player who chose the direction of the story, with typed commands such as “go north” or “kill the goblin”. Nowadays these games are often called *interactive fiction*. EPOC has few games of this type: Castle from Widget Software and Grave Robber by Andy Payndz are examples. Graphical adventure games like Darren Prescott's Castle III have been more numerous and popular on the EPOC platform.

But there is a way that EPOC gamers can have access to a wider range of text adventures: the *adventure interpreter*. This is a native EPOC program that reads the data files of adventures released for other platforms, like the PC. Some companies released series of these adventure games, all using the same format for their data, so a typical adventure interpreter can allow play of a large number of games. I will concentrate on one such program in this article: *FrotzS5*.

FrotzS5 is an interpreter for playing Infocom adventures, one of the more well known being *Zork*. The interpreter is easy to install, being in the form of two SIS files. These are installed in the normal way. FrotzS5 is not document-based, so you can't load an adventure by clicking on it in the system screen. Unfortunately, you can't load a saved game in that way either: in both cases you must load FrotzS5 from the extras bar, and load the adventure or saved game from within the program.

To get any fun from FrotzS5, you need to download an adventure game file. A good place to start is the site *http://ifarchive.org/* which has its own downloadable files, and links to other interesting sites. The adventures for FrotzS5 are in the *if-archive/games/zcode* directory of the site, and have extensions *.z1* through *.z8*. The FrotzS5 documentation states that some games with the extension *.z6* rely on graphics, and may not work. To run a game, simply download it somewhere onto your EPOC machine. Then from the FrotzS5 main screen, select Open Story, and locate the game file with the file browser. The adventure's introductory text will appear and you're ready to play.

FrotzS5 is a very simple program to use, though it has a few features to ensure that reading the screen is comfortable. You can turn bold on and off, show or hide the toolbar, and change the text size using the zoom buttons. The display is automatically configured to fit in the size of your computer's screen, so FrotzS5 runs very well on anything from the Osaris to the netBook.

There are other interpreters: *Hugo*, *Magnetic* and *TADS* being available for EPOC. However, these other interpreters lack FrotzS5's ease of installation and use, being straight ports of desktop software that do not use the EPOC interface. In fact, I had much trouble getting any of them to play a game reliably, so I would recommend them only to those find wrestling with software an entertaining adventure in itself.

It seems that the appeal of this type of game will be timeless. While more graphical games have pushed the text adventure out of fashion in mainstream games, no graphics can improve upon those images that form themselves in the imagination of the player, upon reading a well-written description of the places and creatures seen in a typical adventure game.

Program	FrotzS5
Author	Frédéric Bouvry
URL	www.tucows.com/preview/11282

The Right Connections

The last in the head to head comparison series of Connect 4 games, by Damian Walker

It seems a little unfair to put Puissance4 in a head-to-head with the other programs in this series, because the program is self-confessedly unfinished. As well as a very rudimentary feature set, the game suffers from a lack of artificial intelligence. There is a computer player, but it moves randomly and makes no attempt to block your rows of four.

As it stands, the program offers simple graphics, but no sound. It plays a nearly standard game: 7 columns by 5 rows, rather than the more usual 6 rows. There are virtually no other features: the game works, but is unplayable without a human opponent due to the lack of a coherent computer player.

So why cover it here? Firstly, for completeness—it does exist, after all. Secondly, I wanted to highlight its existence for the very good reason that it's supplied with source code. So anyone familiar with OPL can use this game as a starting point for a more complete version. For that reason alone, it's worth a look.

Conclusion

This has been a very mixed bag of programs to review, even though they all play the same game. While each program had its own merits, two stand out from the others: Power4 and Four In A Line. Power4 has its interesting rules variations, which allow to create what amounts to a different game. But the professional presentation of Four In A Line, and its extensive feature set. Other games have their own things to offer: 4Play with its colour graphics, Connect 4's use of audio, and Puissance 4 with its source

Rank	Game	URL
1st	Four In A Line	http://zingmagic.thezpace.com/home/zview///5
2nd	Power4	http://dcabuzel.free.fr/
3rd	Connect 4	http://www.bjv.btinternet.co.uk/
4th	4Play	http://www.psionplace.com/software/series5/4Play-2001-7-11-psion-series5.html
5th	Puissance4	http://dcabuzel.free.fr/

	4Play	Connect4	4InALine	Power4	Puissance4
2+ lines to win				☆	
Difficulty levels	3	4	3	4	
Help		Dialog	Full	Full	
Hint			☆		
Running score				☆	
Save game			☆		
Starting player	☆	☆			☆
Take back move			☆		
Timer			☆		