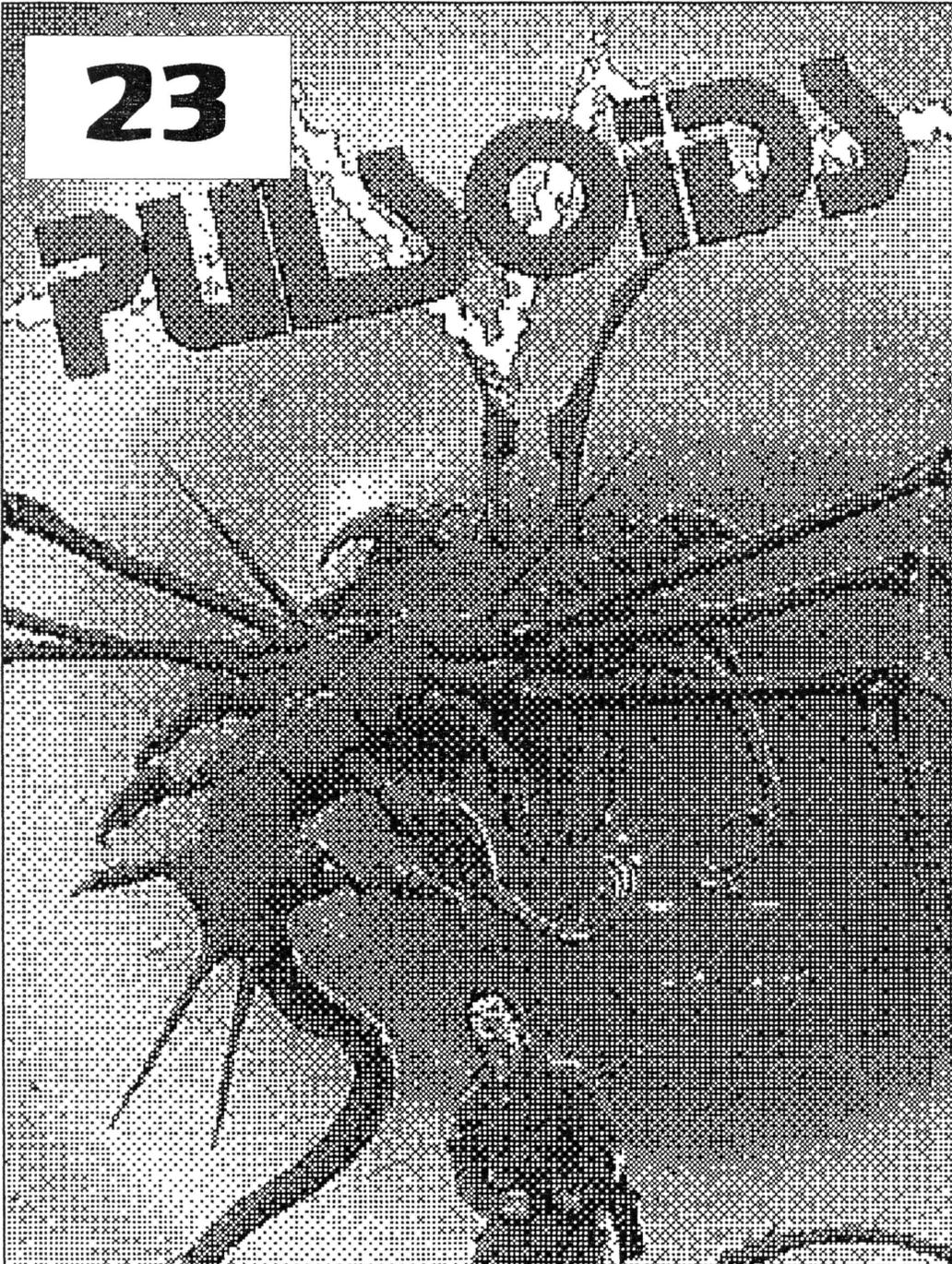


# Rhetoric

23



NEW ORIC GAME!

Twilichte's PULSODS previewed  
Assembly 2002  
Evolution Basic  
Sausages dissed

## CONTENTS PAGE

### 3. ASSEMBLY 2002

'Our Jon' reports on his trip to Finland.

### 6. EVOLUTION BASIC MANUAL

Essential info for this high level language

### 12. PULSOIDS PREVIEW

It's arrived, and you can check out the pics on the new game here

### 13. COMP.SYS.ORIC

Newsgroup nonsense!

### 14. ORIC INVADERS

Classic arcade game to type in, in machine code. More messy hex than you can shake a stick at.

### 16. MICROWAVES

Useful Oric snippets and routines stolen from the old magazines on the top of Chaos's wardrobe.

### 19. ORIC LISP

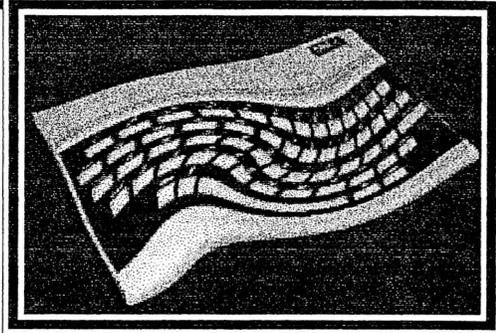
Final installment of the LISP manual.

### 21. BRIAN'S POSER PAGES

Although these pages are called POSER PAGES, strangely, they don't contain any pictures of POISON, TIGERTAILZ or MÖTLEY CRÛE, but they DO have lots of difficult puzzles to frizzle your brain.

### 23. THE GHOST PAGE

Yes .. This page is a special limited edition ghost page, completely invisible to the naked eye.



# RHETORIC ISSUE 23

## GREETINGS...

I don't believe anyone actually reads this part, and after a quick glance at the contents, you'll all be halfway through ASSEMBLY 2002 by now, so I'm going to say a few rude words.... JAHOOBIES, BIG JOBS, and BEARDED CLAM. The first person to write in and complain about the use of bad language (other than Grammar, any ex-teachers out there ) will win a prize. Anyway, apologies for the lateness of this issue. Been a bit ill recently, as well as going on holiday to Cleethorpes (very sick), and having to M.O.T. the Morris Marina. Excuses eh? Hopefully I'll have caught up by the time the final release of PULSOIDS arrives. Injecting a new lease of life into the recently software starved Oric, this game promises a lot, and delivers too, with new standards set in colour use, gameplay and sound. Sorry for no new news about the Oric meet... Should have more info soon, and in plenty of time for you to make your arrangements.

Can I make a final request for you to send in anything for the last 2 remaining magazines. This may be your last chance to make your mark on the 'ORIC PRESS'. Let's make issue 24, and especially the final ISSUE 25, something to remember.

SIMON (CHAOS)  
32 Peter Paine Close, Butterwick, Boston, Lincs. PE22 0HA, UK  
Chaosmongers@yahoo.com

© 2002 RHETORIC

This coupon entitles you to a free ORIC ATMOS from COMET or DIXONS. Simply present to the sales assistant and as for your free computer. (subject to availability)

# Assembly 2002

## *Full Report by Twilight of Defence Force*

I set off on Monday at 4 in the morning to get to my 8:40 flight from Stansted on time. I wasn't going to have a repeat of the VIP4 fiasco (I missed my flight back from Paris and had to pay a whopping 150 quid (It had originally been 42 quid return).

The flight was on time and I transferred flights in Amsterdam. I arrived at Helsinki Airport at 15:30 and after waiting a couple of minutes at the entrance, met Raul.

Raul Hakli is the only Orician I know of in Finland. He put me up last January when I went to Alt Party. He is like Euphoric, the only but the best.

We travelled to Itakeskus, and to his old flat where I was to sleep. I dumped my stuff there, then spent the rest of the day and the following two with Raul and his wonderful family.

Thursday soon came around, and I set off for Hartwall Arena where Assembly was to be held. Hartwall Arena as an enormous stadium, built primarily for ice hockey championships but is used for various purposes during the few short summer months.

On arrival by number 57 bus, I approached this massive building. In the back of my mind was the thought that this place was either the wrong venue or the party was going to only use a small fraction of the Arena.

After spending a few minutes, walking round the circular arena, I found two distinct queues of people waiting to enter.

They were adorned with suitcases, duffel-bags, trolleys and backpacks stuffed full with PC's and their paraphernalia.

Strangely, I saw no other computers in the queue. The people queuing were of various ages, most in their teens or early twenties. I met and talked to a few nameless faces in the crowd, none of whom I recognised. A large proportion of them had travelled across from Estonia, but there was no shortage of Fins, Swedes, Germans and Americans.

We shuffled into the arena around 12 midday, after alternating between the two queues when we noticed crowds drifting towards each one. All our bags were checked rigorously for strange objects (incendiary devices, knives, guns, Spectrums etc) and then I was directed to a ticket stand where I was able to get my Green Official VIP Old-school wrist band. I was now an official part of this great party. After a few trips around, I soon realised that the whole Arena was assigned to this event. Simply thousands of computers were set up in the main stadium, on almost all the stands around the centre stage, on the main ice-rink (Though not iced fortunately), in various halls off from the main arena and in the corridor that followed the full circle of the building. This corridor also had three restaurants (Two Pizza Huts and one Hes-Burger (Finland's version of Macdonald's)) and various sponsors stands.

I managed to find the old-school area which had been shuffled away to a long room in the basement. Old-School is the Europeans classification of people who still enjoy coding and writing demo's for pre 90's computers. Old-school is respected in Europe since they see it as their history, and sometimes more important than the new stuff. However, Assembly organisers had decided that they should put any Coders into this room, that included PC coders as well. I was not too happy about this since it was a dark dank area that many visitors would not even notice when coming to Assembly. It was then that I realised that over 90% of Assembly party goers were Gamers (Main Arena and corridors).

I finally found my allotted E2/12 table and set up my kit. This consisted of my laptop, Music keyboard, Oric (But just the Oric, no monitor) and mug.

So priorities, priorities...

Make a cup of tea!

Since we had been shuffled into the long room and classified as VIP, we had been given a lounge area (Which most of the time, people were sitting with their laptops or sleeping on) and a Drinks stand. Which fortunately had a kettle and those little Milk tippers. After making a nice cup of English Tea (I always take teabags with me), I set off to look for Setok.

Setok was the guy who had organised Alt Party in January, he was also a member of the Demo-group Aggression, famous for their Atari ST Brain Damage Demo. Setok had moved over to Finland from Ireland with his family when he was 12. He had not lost the Irish accent at all.

I managed to find him and I met the rest of his group all except Teque who didn't appear at all at Assembly.

I then went for walks looking at the various sponsor stands (Nokia, AMD processors, Cisco Systems etc) and joined up with Setok again for some food and drink. Unlike mainstream restaurants, the Hartwall Arena's Pizza hut was just another fast food chain, with pizza that was just as nutritious and satisfying as a Macdonald's. So we went for a Hes-burger, which was slightly better but still did not fill the English stomach.

By now it was around 22:00 on Thursday night. I went back to the old-school area to try and find some other computers, than the multitude of humming PC's. I discovered about 5 people with Commodore 64's and watched their demo's for a while. It was now quite late (Around 2AM) and to wake myself up a bit, I went for a walk around the main circular corridor. There I found a stand advertising AMD processors in the form of two computers set up with a racing game and Grand Theft Auto 3. I watched and played Grand Theft Auto 3 for 3 hours before returning to Old-school to meet up again with Setok and Wistom (Technical guy from Aggression). We talked for hours, walking a little and sitting to watch the various demos, and games on show.

At around 5AM, I was shattered, and went back to my set-up and promptly fell asleep for a few hours.

Friday was just as hectic. The legendary Rob Hubbard was due to appear at two seminars Assembly had organised. One at Friday lunchtime, one in the afternoon.

All registering for these seminars had to be done through the internal network. If you had a PC, it was assumed you would also have Ethernet access. Something I did not have though. So I missed out on the first seminar (Golden Days of Computer Music) but attempted to register for the second at 16:00, only to discover it was overbooked. However, there was no-one to take our names at the door so I managed to creep in on the second seminar with Setok, Wistom and another Aggression member. Rob has lost the Beard that I remembered from 80's photos and looked just a tad older and thinner.

He spoke about EA, the company he had worked for and had come to Assembly representing. However, it appears that Rob is no longer with EA and avoided saying what he was going to do now. Primarily though, he spoke about the future of Computer music. It was mildly interesting, if not a little boring at times. Yes!, Rob Hubbard can be boring! And since I'd had just 3 hours sleep since Thursday morning, I kept dropping off.

When the hours seminar was almost over, he asked for questions. I had a bundle of questions myself but felt they were not suitable for the subject matter. And therefore, the seminar ended and we all shuffled out of the room. However, I waited patiently outside for Rob, together with 101 other people bearing Commodore 64's for him to sign. And then I got a crazy idea. I rushed away from the bustling crowd and searched through my suitcase. I returned to the Seminar entrance bearing my beloved Oric Atmos!

After 101 C64's had been signed, plus a game-boy (!?!), a back-plate to a 1541 disc drive and a few Assembly posters, I got my chance to speak to the "God" of Computer music.

My legs were like jelly, my voice completely off-key, but I resolutely introduced myself and presented my Oric for his autograph. Unlike the other fans with their mundane 64's, He suddenly sprang into life and clasped the Oric like a new machine and immediately asked me what sound chip it had! Wow! This was fantastic, I managed to blurt 8912 which he acknowledged and then excused the sound chip by saying that we (I should have said I) have got 3 channels Sample Tracking out of it.

We then went through some of the other spec and he signed it on the back. I managed to get a few more lines in (I asked him if he knew Andrew Moore, author of Defence Force, but he didn't). Silly question maybe but my mind was a complete flurry. I then left him alone to the other fans (They had gradually been accumulating again with their C64's). I saw Setok again with his Video camera (Which he had borrowed from the party organisers to do an Alt version of the proceedings). I asked him why he hadn't taken shots during the seminar. And he had said that there was a chance he was going to be able to have an Alt interview with Rob. He (And I) wanted so much to interview Rob and ask him some of the less pursuing questions. Such as "Does he like the stardom bestowed upon him", and "Does he wish they'd leave him alone now he's been out of the scene for over a

decade".

We managed to ask him, and he kind of accepted so no photos were taken that night, except a small bit of video footage for later use. The tiredness soon returned afterwards as I set off brandishing a "New" Oric Atmos with Rob (The god) Hubbards signature on the back.

I spent the rest of Friday walking around with Setok, trying to find people other than gamers for the ALT TV slot he had been given by the organisers for midnight. We found a couple in the main arena, but distinguishing the multi-skinned IRC and ICQ users from Coder type screens was very difficult.

We watched and filmed a few demos, most notably the C64 demos, and walked around the corridors to find something interesting. We found some Amiga coders who then began to run away. They had organised a small treat of their own. three 10Mb Hard discs were brandished, and apparently it was a famed game to throw them as far as one could. They didn't do this within Hartwall, but walked to a lay-by. We filmed their strange activities and chatted about a new Amiga that was due to arrive from America shortly.

Me and Setok then spent most of early Saturday morning editing the film we had gathered on Wistoms Mac. We then rushed the finished program to assembly film-crew headquarters just having time to return to the Pizza Hut to eat some more trash and watch the half hour Alt TV shoot.

At 9:30 on Saturday morning, I managed to slip away and back to Raul's Flat for a few more hours sleep. I then returned to Assembly for the evenings events.

On Saturday afternoon, I caught the bus back to Assembly, and watched the prize ceremony take place for various compos. Unfortunately, Setok hadn't managed to sort out an interview with Rob Hubbard, nor any real footage of him. This was a great shame since I was planning on gate-crashing a little to get some photos in.

At midnight, I managed to find someone who was heading back past Itakeskus, and after watching some fantastic Wild Compos, I returned with them to Rauls arriving at his flat at 2am

I then went straight to bed, waking at midday (Sunday). I went over to Rauls, and me and his family went out to a local restaurant for a meal. At around 19:00 I rang Setok (I awoke him) and we arranged to meet in the evening at Molly Molones (An Irish pub in the centre of Helsinki). At 22:30 I rang him again and woke him up again, but finally he arrived at Molly's.

We had a great night out, and we talked about almost everything. I finally returned to the flat at 3 am, getting up again at 8 and travelling back to Britain.

**Notes:**

Food was very expensive in Finland, but possibly the same as London prices. However, Assembly prices at Pizza Hut and HesBurger were twice that of anywhere else in Finland. A single piece (1/8) of pizza, Fries and a coke cost 6 Euros (Around £4.50).

I organised my travel arrangements through expedia.co.uk and it cost me just 125 pounds from Stansted (Return + Transfers + Taxes). This was going via Amsterdam, although I now wish I'd booked the GB>>Holland through Easy-jet, since that would have been both a lot cheaper and very much easier for me. Instead it cost me an extra 30 pounds to get to and from Stansted.

Probably the most important thing to take with you (Apart from Passport and Tickets) has to be a sleeping bag. I didn't take one which was a big mistake.

Only after the party did I realise that the VIP status wrist band had allowed me to partake in the customary Sauna on the Friday night with other VIP ticket holders.

Most of the demos had to be registered before Assembly started in order for one to enter each compo. Therefore, I promptly forgot to do so, so missed out on winning the rather generous prize of 3000 Euro for the Best Old-school Music Compo

During the Party, I only recognised one person from the French parties, namely Florent (Or Flod as he is sometimes known), a French guy who I'd met at VIP4.

**Data Dictionary**

**Setok**

Christopher Lawson of Aggression. A Demo Coding Group who produced a demo on the Atari ST called Brain Damage (Very famous)

**Compo**

A Competition

**Hartwall Arena**

The Enclosed Stadium/Arena where Assembly was held.

**Old-school**

A term used to describe coders, graphic artists and musicians who still enjoy retro-computing and programming

**HesBurger**

Finland's version of the Fast food chain Macdonald's.

**Rob Hubbard**

The most famous C64 Musician who wrote literally hundreds of tunes for popular C64 games such as IK+, Parallax, Sanxion, Thrust, Mega-Apocalypse and Commando.

**Alt Party**

A Finnish Party held every January in Helsinki. Quite a bit smaller than Assembly (Who it was estimated 5000 people were there!) and more directed towards old-school and

alternate open minded sceners.

**Sceners**

All Those who go to Parties such as Assembly, VIP, LTP and Alt Parties.

**Assembly**

The name of the Demo-Party in Finland

**Demo-Party**

It is important to mention it here, since the definition seems to be shifting from "A meeting place for people who write demo's" to "A meeting place for people who enjoy using computers to code, play, communicate and express their artistic flair's on".

**CUMBERLAND  
SAUSAGES ARE SATAN'S  
SAUSAGES**



Yes, it's true. Buy Cumberland sausages, and be eternally damned. In recent tests, these sausages were found to contain amongst other ingredients...

- 45% Squirrels feet
- 20% fish eyelids
- 15% Ground Worms in aspic
- 5% Snouts and entrails
- 5% Chloro-flouorocarbons, extracted from old fridges
- 3% Polystyrene (and other members from 70s punk band X-Ray Spex)

WE BELIEVE THAT THE PEOPLE EATING THESE SAUSAGES ARE 90% MORE LIKELY TO RESULT IN BEHAVIOUR SUCH AS BIZARRE BEARD GROWTH, LISTENING TO DEEP PURPLE RECORDS BACKWARDS, AND WORSHIPPING THE DEVIL.

**DON'T TAKE THE RISK...**

Issued by Lincolnshire County Councils Sausage Marketing Board.

**ORIC MEET  
UPDATE**

Date hasn't yet been agreed, but  
Looking like it's going to go ahead.  
More info when I get it!

# EVOLUTION BASIC MANUAL

## Introduction

Evolution Basic is both a Basic you are familiar with (because it is nearly 100% upward-compatible with Oric Basic 1.1), and at the same time, a highly enhanced language with advanced structured concepts that allow to build big programs in an easier way.

Evolution Basic is not a Basic extension that adds a few utility commands to Tangerine's work, such as lines-renumbering or the like; instead it introduces major changes in the core of Basic 1.1 (as you might know, this core was written by Microsoft). Evolution Basic aims to be a direct rom-replacement for modern Oric users: as its name suggests, it allows you to enter a new world of home computing and still be able to run your old Oric programs (Basic and non-Basic ones). However, if you are a Basic developer with a tape-only Oric configuration, you are out of luck because tape-saving routines have been removed in Basic Evolution (tape-loading routines are still there so you can load your old programs). For this reason, Evolution Basic is ideally used by Basic developers with a disc configuration.

I think the advanced features of Evolution Basic are worth the efforts of abandoning bad habits: without resorting to a totally new language, you will be able to write bigger programs in a cleaner and easier way as it removes most of the limitations of Basic 1.1.

The changes are detailed in next paragraphs, have a nice reading and happy programming !

Fabrice

## 1. Variable names, and Basic keywords

In Basic 1.1 and earlier, names are recognized up to their second character, so that TEMPORARY and TEST actually denote the same variable: TE. For this reason, programmers have the habit to use one- or two-characters variable names: this makes programs harder to read because you quickly don't remember what this NX variable is used for...

Evolution Basic removes this limitation: a name still starts with a letter, and is eventually followed by letters and/or numbers, but...

### **Change # 1: Every character of a name is significant.**

So, you can have TEMPORARY and TEMPERATURE variables, they will be different.

A name can be up to 127 characters (63 characters in the case of array names). This allows you to more easily choose names for your variables, and more importantly, be able to guess what a program does when reading it.

Another big hassle I experienced in Oric Basic was that it didn't allowed Basic tokens (keywords) in names. For example, you couldn't use names like SORT or FUSION because they respectively contain Basic keywords OR and ON. This is due to an over-simplistic token recognizer, so it has been completely rewritten inside Evolution Basic.

### **Change #2: Names may contain Basic tokens.**

The old Basic allowed to write poorly readable statements such as

```
IFSORTTHENAME=BANDIT
```

Unless you write Oric Basic programs every day, would you correctly read the above statement as

```
IF S OR T THEN AME = B AND IT ?
```

As you can see, spaces largely enhance the readability of a program, but Basic programmers don't use them for the reason they take... space.

Evolution Basic behaves in a very different way: because of Change #2, you have to enter spaces in order to separate consecutive names in a program (otherwise the whole sequence would be considered as a single name: remember that a name ends at the first non-alphanumeric character), but don't worry, they won't consume memory (they are not stored).

Thus, writing IFSORTTHENAME=BANDIT is an assignment of variable IFSORTTHENAME with the value of variable BANDIT, whilst IF S OR T THEN AME = B AND IT is the expected conditional statement.

This is the most visible change of Basic Evolution because you will get a SYNTAX ERROR when typing LIST50- (you have to insert a space after LIST: LIST 50- ). In the other hand, this allows you to use variable names like LIST1 or LIST50.

Note that the LIST routine has been rewritten too: it automatically displays spaces when two names (or Basic keywords) are in sequence.

To summarize:

### Change #3: Spaces are removed on entry, and added when listing.

One exception that allows further enhanced readability is called indentation: you can add extra spaces at the very beginning of a line in order to emphasize the program structure (e.g.: lines inside a FOR...NEXT loop).

### Change #4: Indentation spaces are kept inside the program.

## 2. Control structures

Let's start with a small quizz...

Using Basic 1.1, what is the result of the following statement ?

```
IF 1<2 THEN A=1:PRINT"TRUE" ELSE A=0:PRINT"FALSE"
```

Easy ? Try it and you will see:

```
TRUE  
FALSE
```

This is because Basic 1.1 only skips one instruction in the ELSE part when the condition is true.

Another one ?

```
IF 1>2 THEN IF 0=0 THEN PRINT"A" ELSE PRINT"B" ELSE PRINT"C"
```

An Oric-1 displays B, and an Atmos displays nothing !!

To understand why both versions are erroneous, you have to know that the Microsoft core didn't include an ELSE statement, this was added by Tangerine in the Oric-1 version. It was largely wrong in Oric Basic 1.0 and not much better in Oric Basic 1.1. In fact, only simple conditional statements (with only one statement following the ELSE part) are handled correctly.

Evolution Basic keeps Basic 1.1's IF statement (for compatibility issues), but...

### Change #5: A multi-lines form of IF statement is added :

```
IF A<B  
THEN A=1:PRINT"TRUE"  
    REM SEVERAL LINES CAN BE WRITTEN  
    REM IN THE "THEN" PART  
ELSE  
    REM AND IN THE "ELSE" PART TOO  
A=0:PRINT"FALSE"  
END IF ' THIS IS WHAT MARKS THE END OF THE IF STATEMENT
```

With this new syntax, the THEN keyword is written on the second line, this is what differentiates the two accepted forms. Of course, the new IF allows to nest several conditional statements with a correct behavior.

### Change #6: Evolution Basic offers a WHILE ... END WHILE control structure :

```
WHILE A>0 AND B>0  
    IF A<B  
    THEN B = B - A  
    ELSE A = A - B  
    END IF  
END WHILE
```

This form of loop is more general than a REPEAT ... UNTIL : it allows to build loops that are executed any number of times, including zero times (the statements inside a REPEAT ... UNTIL loop are always executed at least once). Note that the condition of the WHILE indicates the case where the inner statements have to be executed, whilst the condition of the UNTIL indicates the case where the processor has to exit from the loop.

There is no limitation to the number of statements inside the THEN or ELSE part, or between WHILE and END WHILE. Thus, in order to speed up the jump to the ELSE part (when the condition is false) or to an END IF or END WHILE, **the following keywords must be written in first position on their respective lines (after any indentation spaces): WHILE, THEN, ELSE, END IF, END WHILE.**

This should not be a heavy constraint as this also allows to better show the structure of a program, with a good indentation.

## Subroutines and local variables

Beyond control structures, subroutines are the key to structured programming. They allow to follow the functional decomposition issued from the analysis, and to lower the complexity of a program by splitting problems in sub-problems. When writing an algorithm that solves a sub-problem, you don't need to have the initial problem or other sub-problems in mind: sub-problems are solved independently.

But... this is not possible with Oric Basic 1.1 because subroutines (called by GOSUB) are not independent at all: the only way to give information to a subroutine is through variables of the main program, and when writing a subroutine with Basic 1.1, you must take care of the variables you use in the subroutine: if they are also used in the main program (or in another subroutine), you get strange side-effects. In other terms, the variables you use in a subroutine are variables of the main program, and a Basic 1.1 subroutine is nothing more than a set of lines that has been cut off from the main program in order to be used several times.

### **Change #7: Evolution Basic introduces a new kind of subroutines: they are named, and called with the new SUBR keyword.**

```
E.g.:  DEF SUBR MENU
        PRINT "MENU"
        PRINT "1- LA SOUPE : VARIABLE NAMES"
        PRINT "2- LES ENTREES : CONTROL STRUCTURES"
        PRINT "3- LE PLAT DE RESISTANCE : SUBROUTINES"
        RETURN
```

This subroutine is called by a SUBR MENU statement. Simple, isn't it ?

### **Change #8: Evolution Basic's subroutines accept parameters.**

```
E.g.:  X1=20 : Y1=13 ' CENTER OF SCREEN
        FOR X=5 TO 10 ' HALF HEIGHT OF BOX
            GET CH$
            SUBR BOX (X1-X, Y1-X, X1+X, Y1+X, CH$) ' DRAW A CENTERED BOX
        NEXT
        END
        '
        DEF SUBR BOX (X1, Y1, X2, Y2, CH$)
        FOR X=X1 TO X2
            PLOT X, Y1, CH$ ' TOP SIDE
            PLOT X, Y2, CH$ ' BOTTOM SIDE
        NEXT
        FOR Y=Y1 TO Y2
            PLOT X1, Y, CH$ ' LEFT SIDE
            PLOT X2, Y, CH$ ' RIGHT SIDE
        NEXT
        RETURN
```

If you are familiar with Oric Basic, the above program will make you protest: "Beware, variables X, X1 and Y1 are used by the subroutines, your program will go berzek !". No, don't worry, this is no more true, because...

### **Change #9: All the variables in a SUBR are local to this subroutine; parameters variables are local too.**

This means there can't be any interference between variables of a subroutine and variables of the main program (or variables of another subroutine): subroutines work with their own set of variables.

Let's give some details about the work done by the SUBR BOX(X1-X, Y1-X, X1+X, Y1+X, CH\$) call:

- a definition of SUBR BOX is looked for, through the whole program code.
- variable X1 is created, even if it already exists in the main program. It is a second X1 variable, that will disappear when the subroutine returns. This new X1 variable is initialized with value X1-X, where X1 is the variable pertaining to the main program (so, at the first call of SUBR BOX, value is 20 - 5 = 15). The other parameter variables (Y1, X2, Y2, CH\$) are created in the same way. If the number of arguments were different than the number of parameters in the definition, a "BAD PARAMETERS COUNT ERROR" would occur. If types (numeric or string) did not match, of course, a "TYPE MISMATCH ERROR" would be raised.
- once the subroutine is entered, the variables of the main program disappear ! From this time on, only the subroutines' variables exist.

- each time a new variable is used (for example, loop variables X and Y), it is created as usual (there's no need to declare variables in Basic) and it is local, so...
- when RETURN is reached, all the local variables of the subroutine are suppressed (the memory storage is reclaimed) and the variables of the main program re-appear (they were hidden during the subroutine execution). Did you say "It's magic!" ?

If these concepts of subroutine parameters and local variables are totally new for you, breath a while before reading the rest...

## Change #10: Evolution Basic parameters are bi-directional ! You can return results through parameters.

The mechanism used for parameters passing is the same as the one found in Fortran: when calling a subroutine, if you provide a variable for the argument of the call, its value is of course passed to the parameter variable during the call, but... it will also be updated with the value of the parameter variable when the subroutine returns.

Here is a simple example:

```
' MAIN PROGRAM
A=3 : B=5
SUBR ADD(A, B, C)
PRINT C
END
' SUBROUTINE ADD
DEF SUBR ADD(X, Y, Z) ' Z EST UN PARAMETRE RESULTAT
Z=X+Y
RETURN
```

When calling subroutine ADD, the X,Y,Z parameters are passed the respective values 3, 5 and 0. When the subroutine returns, variables A, B, C are updated with values 3, 5 and 8.

Of course, this update can only be done if a variable is provided as argument. When you write a subroutine, you have to define what parameters the subroutine needs and what results it returns. Actually, three parameters passing modes are implicitly available (instead of being explicitly specified as in other languages like Ada): a parameter can be an input parameter, an output parameter, or both an input and output parameter. If you want to ease re-utilization of your subroutines by other people, you should document the usage of parameters and be respectful to their passing mode, i.e. use input parameters in read-only operations.

This powerful subroutine mechanism opens the door to a library of subroutines for Evolution Basic. Ok, here is a first one for the collection:

```
DEF SUBR BINOMIAL(A, B, C, N, R1, R2)
' COMPUTES THE ROOTS OF A SECOND DEGREE EQUATION
' A*X^2+B*X+C=0
' A, B AND C ARE THE COEFFICIENTS OF THE BINOMIAL
' THE SUBROUTINE RETURNS N - THE NUMBER OF REAL ROOTS,
' AND R1, R2 THE ROOT THEMSELVES
,
DELTA=B*B-4*A*C
IF DELTA<0
THEN N=0
ELSE
IF DELTA=0
THEN N=1 : R1=-B/(2*A) : R2=R1
ELSE N=2
R1=(-B-SQR(DELTA))/(2*A)
R2=(-B+SQR(DELTA))/(2*A)
END IF
END IF
RETURN
```

and an example of use of this subroutine could be:

```
SUBR BINOME(1, -2, 1, NB, X1, X2)
IF NB=0 THEN PRINT'' PAS DE RACINES REELLES''
```

```

IF NB=1 THEN PRINT''UNE RACINE DOUBLE : '' ;X1
IF NB=2 THEN PRINT''DEUX RACINES : '' ;X1,X2
END

```

Well, if you try all the examples given in this section, parameter passing should soon have no secrets for you... However, there's an other way to exchange information between a program and a subroutine in Fortran, through the use of a "common" zone, which is a mean of sharing variables of the main program and of a subroutine. It is somewhat different in Evolution Basic:

### **Change #11 : In Evolution Basic, all the arrays are shared by the main program and the subroutines.**

This is not as clean as in Fortran: you have to use the same names for the arrays in the main program and in the subroutines. Actually, the arrays are not affected by the subroutine mechanism: there is no local array, arrays belong to the main program, even if they can be used in subroutines.

Time for a break, before introducing the last concept related to Evolution Basic's subroutines...

Here we go... nobody prevents you to call a subroutine from inside another subroutine: during the call, local variables of the calling subroutine disappear and only the local variables of the called subroutine are accessible. When the called subroutine returns, its local variables are removed, and the local variables of the caller subroutine are back again. Evolution Basic's subroutines even allow recursivity, i.e. a subroutine can call itself. Beware in this case not to abuse of this powerful mechanism: the 6502 processor stack is very small, so you won't be able to recurse deeply (around 20 calls max., and this number will be lowered again if you nest control structures). If you recurse too deeply, you will get a "MEMORY OVERFLOW ERROR". Hopefully, Basic is more efficient with iterative programming than with recursive programming, so you will usually choose an iterative style than a recursive one, and thus avoid this stack overflow. However, there is a large family of problems that are more easily solved by a recursive style than by an iterative one: as long as you don't recurse too deeply, you can do it with Evolution Basic. Here is an example of the classic 8-Queens problem: I let you judge of the elegance of this program, of course Oric Basic 1.1 is not able to do it that way...

```

DIM COLUMNS(8)
CLS
PLOT 4,2,"ABCDEFGH"
PLOT 4,11,"ABCDEFGH"
FOR I=1 TO 8
  PLOT 3,11-I,CHR$(48+I)
  PLOT 12,11-I,CHR$(48+I)
NEXT
SUBR QUEENS(8,FALSE)
END

DEF SUBR QUEENS(QUEEN,MENACE)
IF MENACE THEN RETURN
IF QUEEN=0 THEN PRINT@0,14;:STOP:RETURN
COL=8
WHILE COL>0
  MENACE=FALSE:ROW=QUEEN+1
  WHILE ROW<=8 AND NOT MENACE
    IF COLUMNS(ROW)=COL THEN MENACE=TRUE
    IF ABS(COLUMNS(ROW)-COL)=ABS(QUEEN-ROW) THEN MENACE=TRUE
    ROW=ROW+1
  END WHILE
  COLUMNS(QUEEN)=COL
  PLOT 4,QUEEN+2," "
  PLOT COL+3,QUEEN+2,127
  SUBR QUEENS(QUEEN-1,MENACE)
  COL=COL-1
END WHILE
RETURN

```

## 4. Things that have disappeared from Basic 1.1

Space was needed to add all the new concepts in an already full 16 KB rom, so many routines have been optimized and compacted, but others had to be removed. Here are the things that have been removed, they have been chosen for their rare usage

in existing programs.

### **Change #12: STORE and RECALL routines have been removed.**

These keywords were introduced in Basic 1.1 in place of Basic 1.0's INVERSE and NORMAL. These routines are not used a lot, so this shouldn't bring many compatibility issues. The RECALL keyword is still present in Evolution Basic, in order to ease detection of programs that use this command : a SYNTAX ERROR is raised by the command. The keyword STORE has been replaced by WHILE.

### **Change #13: CSAVE is ineffective on real Oric.**

As mentioned in the introduction, Evolution Basic ideally targets Oric users with disc configurations. If you have a tape-only configuration and do not develop programs, you can still install Evolution Basic in your Oric in order to play with its features, but you won't be able to save your programs on tape. The routine responsible for writing a byte to the tape has been removed, so the CSAVE commands seems to work, but actually it doesn't write your programs to the tape. However, due to a trick in Euphoric, you can save programs on .tap files with Euphoric (there's no such trick in Atmosfairy, so CSAVE is ineffective in Atmosfairy, like on the real Oric).

### **Change #14: EDIT command has been replaced by an extern program, line numbers are not needed.**

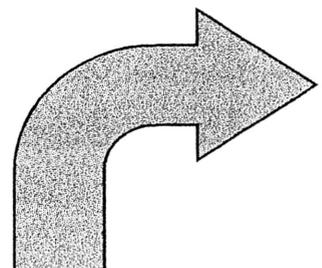
The EDIT command is normally a direct command and does not appear in programs. Also, it was more or less buggy, particularly when the cursor was at the bottom of the screen, and a more-than-40-characters line was asked for editing. So, the keyword has been replaced by SUBR. So, on tape configurations, you have to type LIST 110 instead of EDIT 110 if you want to edit line #110. However, for those using disc configurations, I have written a small full-screen editor: it is named EDIT.

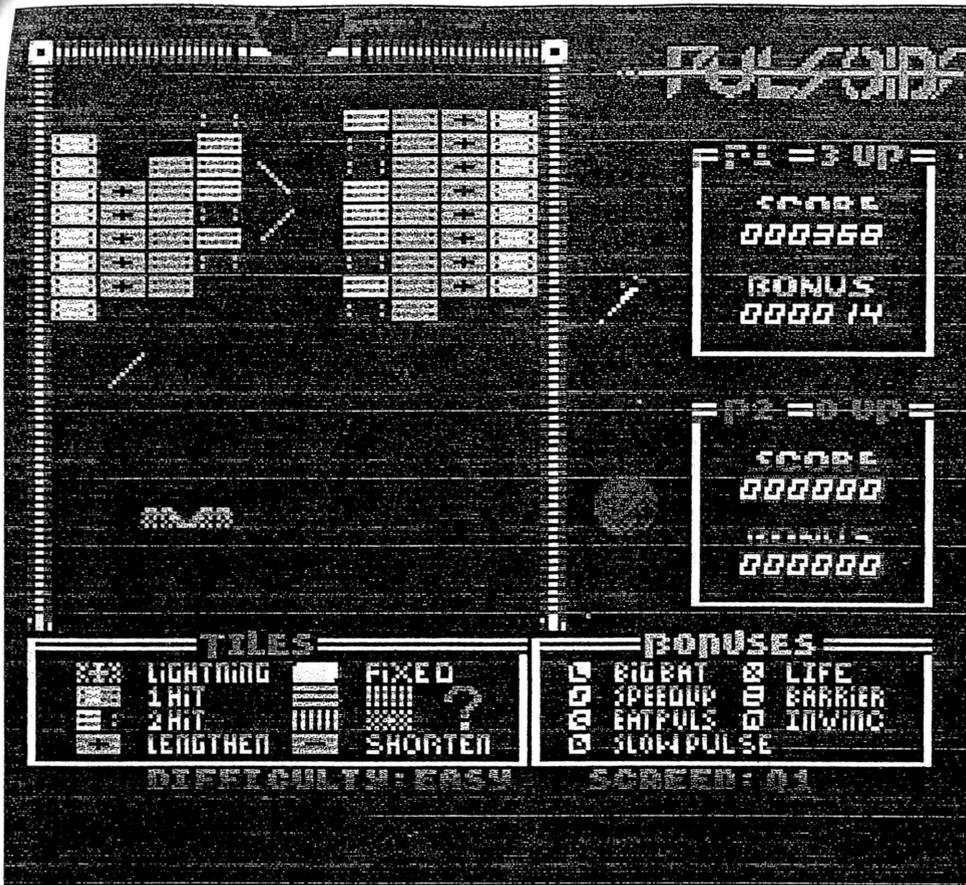
EDIT is an editor for new well-structured programs: I should have mentioned it before, but if you use the new structured-IF and WHILE statements, you shouldn't have the need for any GOTO in your program. And if you use the new subroutines, you shouldn't have any need for line numbers. As a matter of fact, EDIT doesn't use line numbers in the old way : it automatically numbers the lines starting from 1, so a program is handled like in a document editor. Moreover, when you stop a program (Ctrl-C or STOP) and then invoke the editor, the cursor is automatically located on the interrupted line. EDIT doesn't install itself in the memory devoted for Basic programs, so it allows to edit very big programs. It also has other interesting features, be sure to read its own documentation.



OK..... perhaps the biggest and best new game for the Oric is on it's way. It's not just another promise of an idea, but it's arrived. RHETORIC is lucky enough to have received a preview copy of Twilight's new game, which I think will be the biggest game to hit the Oric since Zip N Zap. There's so much to say about it too...

GRAB A BEER, LIGHT A FAG, BRACE YOURSELF and TURN THE PAGE NOW...



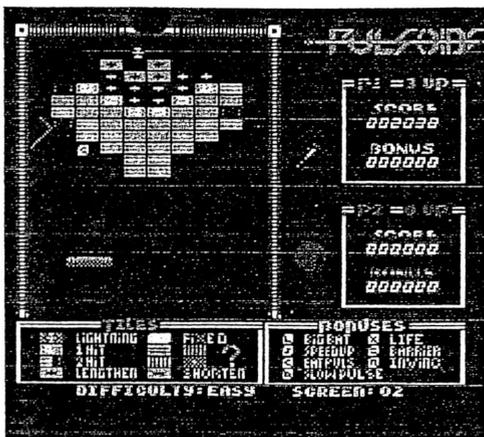


PULSOIDS is converted from a little known game by Mastertronic, which came out on the Commodore 64, Spectrum and Amstrad CPC in the 80s, which itself pays homage to the arcade classic 'ARKANOID'. To the layman, it's BREAKOUT with knobs on. And what a lot of knobs it's got too.

The object of the game is to progress through the levels, obliterating the varying arrangement of blocks (or tiles) on the screen. Some blocks can be easily knocked out, some need more effort, and some can't be destroyed at all. Whilst veterans of Arkanoid (or Breakout if you're old enough) will remember bouncing a ball to hit the blocks, this game has these pulse things which always move diagonally and are deflected off your bat at 45 degree angles. To make life more tricky, there are strange creatures that come through from the top of the screen and hover around. Should your pulse hit one of these, then I splits into two. To add to this, certain blocks/tiles when hit will lengthen or shorten the length of

the pulse. Certain power ups allow your bat to 'eat' the excess lengths of these pulses, in return for a bonus score, payable at the end of the level. Other blocks have other powers, some of which we're not told about.... You'll have to find that out yourself! As you could imagine, with all of these things happening, and pulses lengthening and splitting into two, things start to get pretty hectic, and you'll soon find yourself battling to juggle a dozen really long pulses at the same time! As this is a preview, I'm not going to go into any opinions of this game just yet, as a few finishing touches are planned, and a full and detailed review will appear when the game is officially released. There are plenty of features though, including a hall of shame, as well as a hall of fame, a 2 player option, 4 skill levels (let me just say that level 4 is mind blowingly quick). To whet your appetite, here are some of the features, as well as a planned release schedule. (YES! RHETORIC IS GOING TO GET THE GAME... Yippee!)

- Game Conversion from C64 with enhancements
- Available in French and English
- Available for Oric1, Atmos, Telestrat and Euphoric.
- Full colour (No colour collision) HIRES graphics
- Full 3 channel Sound Effects
- Intro-section
- First use of SID-Sound in game title
- In-game instructions
- 4 difficulty levels, 4 control types (Auto-detect Telestrat Joystick)
- Hall of Shame and Hall of Fame, 10 hiscores in each, both saved/loaded From disc (Auto-detection of disc-drive)



**October 1st**  
 Inform Comp.sys.oric subscribers about the game, together with a link to screenshots of the game in Jede's Software database.  
 Send initial demonstration versions (The latest versions we have) to CEO (Jede) and Rhetoric (Simon Ulliyatt/Chaos) for them to preview.

**November 1st**  
 Send the final version (With disc fix and IJK joystick emulation) to Both CEO (French Version) and Rhetoric (English Version) for inclusion in the December disc.

**December 1st**  
 CEO/Rhetoric release disc to their subscribers with full review in magazine.

**December 14th**  
 Enable free download of game in Software database.  
 Offer limited edition versions of software. This will take the form of 10 sets of Cassette, 3" and 3.5" in specially designed Boxes/Cases.

This means that for now, you should do a small preview of the game, a Full review being in December.

# TIMELINE

# COMP.SYS.ORIC

LATEST SNIPPETS OF NEWS FROM COMP.SYS.  
ORIC NEWSGROUP

I am proud to announce a new release for the ORIC...  
"Meurtres en série" is the new game from Dominique Pessan.  
A Graphic Adventure featuring full Colour "High Resolution" graphics.

For existing CEO subscribers (You lucky devils), you'll get the game before the rest.

To subscribe to the one to offer such exciting releases before the rest, follow this link... <http://ceo.oric.org>

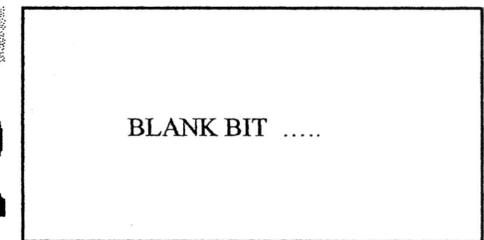
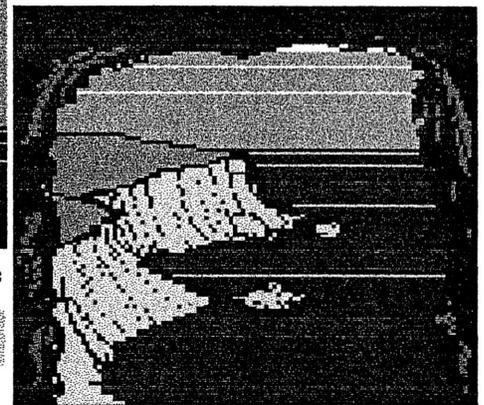
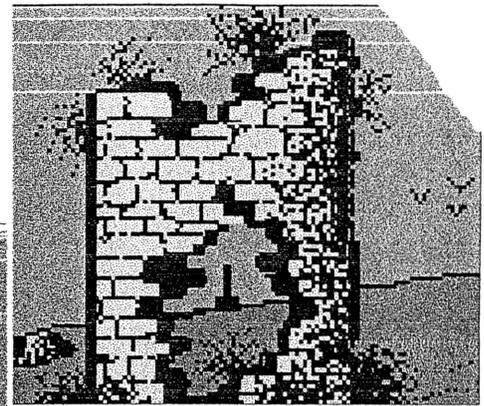
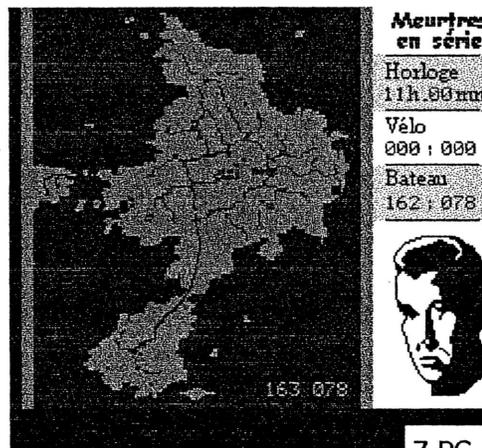
Debrune Jérôme (Correction : Jonathan Bristow)  
<http://www.oric.org>

The staff at Micro Mart want to run a feature over the 8 weeks running up to Christmas and New Year time (I think) which they are calling "The All Time Computer World Cup". The idea is 2 computers from the list below are paired up against each other, and someone who is passionate about a particular format (IE, Vic 20) tells the Micro Mart reader base why it is better than it's opponent (IE, Sinclair ZX81), or why it is generally a truly great computer, and then the readers vote somehow, and the winner goes through.

From this, Micro Mart readers will decide their "Best Ever" computer out of these sixteen; (In no particular order)

All Time Computer World Cup

- 1 Dragon 32
- 2 Commodore 64
- 3 ZX Spectrum
- 4 AMSTRAD PCW
- 5 AMSTRAD CPC
- 6 Atari ST

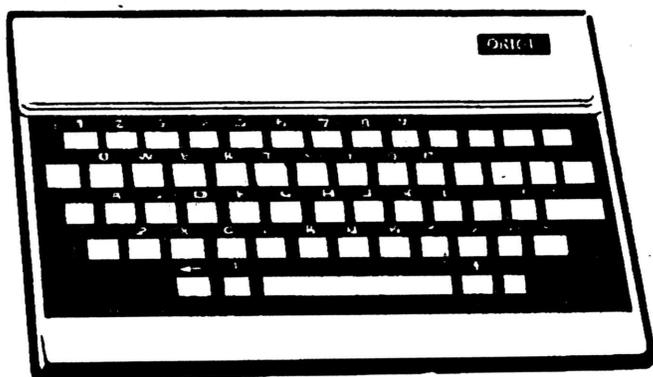


I will need someone who could argue the Orics case etc... Please contact me for more information.

[Shaun@MicroMart.co.uk](mailto:Shaun@MicroMart.co.uk)  
<http://www.micromart.co.uk>

Shaun.

That old favourite space invaders is still as addictive as ever. Denis Salisbury details a derivation of it for the Oric-1 micro.



# ORIC INVADERS

THIS IS A VERSION of Space Invaders written for the 48K Oric 1. The game is played in text mode, using Oric's user-defined characters to create the effect of high resolution.

The program consists of two sections. The machine code must be entered first, using the short Basic routine in listing 2. When all the machine code has been typed in you should type

NEW (RETURN)

to get rid of the small Basic routine.

After typing in all these numbers you may not wish to type in the Basic part of the program — in listing 1 — straight away.

Therefore, if you wish to Save the machine code at this point type:

CSAVE "MC", A# DOO, E# 1403 (RETURN)

Then type:

CLOAD "MC", A# DOO, E# 1403 (RETURN)

when you wish to load it back.

When you have entered the Basic program it is essential that you Save the whole program on cassette before you Run it in case there are any errors in the machine code. To Save the program type:

POKE # 9C, # 1405 (RETURN)

CSAVE "INVADERS", AUTO (RETURN)

This will Save the machine code after the Basic program and will Run the program when it is CLoaded.

The game consists of three different types of invaders, worth 100, 50 and 30 points, and a blue ship which occasionally shoots across the top of the screen worth 500 points. The controls are as follows:

Cursor left — Move left  
Cursor down — Move right  
Space-bar — Fire

(continued on page 125)

Listing 1.

```

0 REM FOR E READ HASH CHARACTER
1 GOSUB 101:GOSUB 700
2 GOSUB 810
3 GOSUB 980
4 GOSUB 490
5 GOSUB 1000
7 END
10 REPEAT
20 READ I
30 POKE 46848+8,I
40 B=8-I
50 UNTIL I=99
90 DATA 8,20,8,0,0,0,0,0
100 DATA 32,32,31,15,15,29,63,63,31,12,15,24,32,32,0,0
110 DATA 1,1,31,60,60,46,63,63,62,12,60,6,1,1,0,0
120 DATA 0,0,0,49,11,7,13,31,59,28,15,23,32,32,0,0
130 DATA 0,0,0,35,52,36,44,62,35,14,60,58,1,1,0,0
140 DATA 0,0,7,13,31,61,6,63,60,27,15,16,32,32,0,0
150 DATA 0,0,36,60,62,47,63,63,15,34,60,2,1,1,0,0
160 DATA 0,3,15,15,15,15,0,12,63,63,63,63,63,0,0
170 DATA 12,12,12,12,12,12,12,14,4,14,14,14,14,14,4
180 DATA 0,15,31,42,53,42,31,15,0,60,62,43,21,43,62,60
190 DATA 63,63,63,63,63,63,63,63
200 DATA 3,15,31,31,63,63,63,48,60,62,62,63,63,63,6
3
210 DATA 63,63,63,51,33,0,0,0,0,0,0,33,51,63,63,63
220 DATA 0,16,40,16,4,13,15,15,15,16,40,16,0,17,59,63,0
18,20,8,20,60
40,60
170 DATA 12,12,12,12,12,12,12,14,4,14,14,14,14,14,4
180 DATA 0,15,31,42,53,42,31,15,0,60,62,43,21,43,62,60
190 DATA 63,63,63,63,63,63,63,63
200 DATA 3,15,31,31,63,63,63,48,60,62,62,63,63,63,6
3
210 DATA 63,63,63,51,33,0,0,0,0,0,0,33,51,63,63,63
220 DATA 0,16,40,16,4,13,15,15,15,16,40,16,0,17,59,63,0
18,20,8,20,60
40,60
230 DATA 0,0,0,0,4,10,4,0,0,24,44,52,36,24,0,0,99
250 RETURN
    
```

```

490 PRINT CHR$(A);CHR$(17)
500 CLS:INK 2:PAPER 0
510 FOR I=0 TO 3
515 PLOT 0,21+I,1
520 PLOT 6+(I*8),21,"ULV"
530 PLOT 6+(I*8),22,"ttt"
540 PLOT 6+(I*8),23,"ttt"
550 PLOT 6+(I*8),24,"t t"
560 NEXT I
565 POKE28FC7,109:POKE28FC8,110:POKE28FC9,111
600 FOR I=0 TO 32 STEP 3
610 FOR J=0 TO 3 STEP 3
620 PLOT I,2+J*H,"ac":PLOT I,3+J*H,"bd"
630 PLOT I,8+J*H,"eg":PLOT I,9+J*H,"fh"
640 NEXT J
650 PLOT I,14+H,"ik":PLOT I,15+H,"jl"
655 NEXT I
660 PLOT 0,26,7
665 PLOT 0,28,7
667 PLOT 0,1,4
670 RETURN
700 B=0
710 REPEAT
720 READ I:POKE 48880+8,I
730 B=B+1
740 UNTIL B=40
750 DATA 32,32,72,73,45,83,67,77,82,67
760 DATA 83,48,48,48,48,48,32,32,66,63
770 DATA 83,48,32,31,32,32,32,83,67
780 DATA 79,82,67,32,48,48,48,48,48,32
800 RETURN
810 POKE48,3
820 FOR I=0 TO 4
830 POKE 44+I,48
840 NEXT I
850 DOKE880,0
    
```

```

855 H=0
860 RETURN
865 POKE25A,50
890 DOKEE5E,CBEZE+H*40
895 POKE40,180
900 POKE61,0
905 POKE6F,0
910 DOKE70,0
920 POKE72,285
925 DOKE73,CBFCB
930 DOKE75,CBFCB
935 DOKE77,CBFD
940 POKE78,0
945 POKE7E,0
947 POKE 0,0
950 POKE 1,50
955 POKE 5,1
960 POKE 6,3
970 RETURN
1000 CALLED00
1010 IF PEEK(46E)=0 THEN 1080
1020 IF PEEK(45A)<0 THEN 1050
1030 IF H=4 THEN 1050
1040 H=H+1
1050 GOSUB 880
1060 GOSUB 500
1070 GOTO 1000
1080 PLOT 6,12,"PRESS RETURN FOR ANOTHER GAME"
1090 REPEAT
1100 UNTIL PEEK(420B)=175
1110 GOSUB 810
1115 H=0
1120 GOSUB 880
1130 GOSUB 500
1140 CALLED100
1150 GOTO 1000
    
```

Listing 2.

```

10 REM FOR E READ HASH CHARACTER
20 PRINT "INPUT START ADDRESS IN HEX"
30 PRINT "(PRECEDED BY A HASH CHARACTER)"
40 INPUT B
50 PRINT "WRITE OR READ (R/W)"
60 INPUT A#
70 IF A#="R" THEN GOTO 135
80 FOR N=8 TO 1403
90 PRINT HEX$(N),
100 INPUT I
110 POKE N,I
120 NEXT
130 STOP
135 PRINT "PRESS ANY KEY"
140 FOR N=8 TO 1403 STEP 10
150 PRINT HEX$(N); " - ";
160 FOR B=0 TO 9
170 PRINT HEX$(PEEK(N+B)); " ";
175 X#KEY#;IF X#="" THEN 175
177 X#=""
180 NEXT
190 PRINT
200 NEXT
    
```

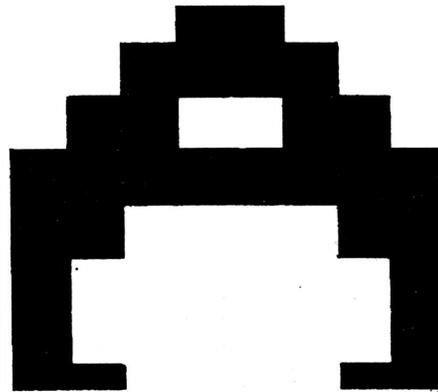
(continued from page 123)

A base is lost if you are hit by an invader's bomb or if an invader reaches the row above your base. The invaders start one row further down after each sheet is completed and they start moving faster the more you shoot.

If you wish to alter the number of bases then change line 810 to:

```
POKE# GE, X
```

where X is the number of bases — 1 to 9 — and change the 51 in line 770 to 48 + X. If the characters I have defined are not to your liking then these are contained in lines 10 to 250 — characters 96 to 126.



Hex dump.

```

ED00 - EA5 E70 EC9 E0 EDO E6 E20 E80
ED0A - E80 E13 EA6 E1 ECA EDO EFD EA5
ED14 - E4 EDO E3 E20 E50 EE EA6 E1
ED1E - EFD EA5 E70 EC9 E8 EDO E25 EAD
ED28 - EC9 EDE EDO E18 EA5 E0 EC9 E0
ED32 - EA9 E1 E85 E0 EA9 E72 EBD ED2
ED3C - E73 EBD E03 EBB EA2 E8D EA0 EFA
ED46 - EFA E20 E80 E12 EA5 E61 EC5 E60
ED50 - EA9 E0 E85 E61 E20 E10 E12 E66
ED5A - E70 EC9 E0 EDO E73 E20 E50 EE
ED64 - ECA EDO EFD E66 E70 EA5 E70 EC9
ED6E - E4 EA9 E0 E85 E70 EA5 E71 EC5
ED78 - E14 EA5 E5A EC9 EA E10 E4 EA9
ED82 - E5 E20 E90 EF E20 E10 E11 EA9
ED8C - E71 E66 E71 EA5 E6E E18 E69 E30
ED96 - E8B EA5 E6E EC9 E0 EFO E9 EA5
EDA0 - E1 EFO E3 E4C E0 ED E60 E63
EDAA - E35 E55 E55 E55 E55 E55 EAD E8
EDB4 - EAC EDO E3 E20 EDO ED EAD E8
EDBE - E84 EDO E3 E20 E0 EE EAD E8
EDC8 - E84 EDO E3 E20 E30 EE E60 E63
EDD2 - EC9 EBD EFO E20 EA9 E8A E85 E77
EDDC - EB1 E77 EC9 E71 EFO E2 EAO E0
EDE6 - EAO E1 EA9 E20 E91 E77 EE6 E77
EDF0 - EC9 EFD EDO E66 E63 E73 E60 E63
EDFA - E35 E55 E55 E55 E55 E55 E73
EE04 - EFO E20 EA9 E80 E85 E77 EAO E0
EE0E - EC9 E71 EFO E2 EAO E1 E91 E77
EE18 - EA9 E20 E91 E77 E66 E77 EA5 E77
EE22 - EDO E66 E66 E73 E60 E63 E55 E55
EE2C - E55 E55 E55 E55 EA5 E7E EFO E1
EE36 - E7E EA5 E73 E38 E9 E28 E85 E75
EE40 - E85 E76 EA9 E70 EAO E0 E91 E75
EE4A - E85 E7F E60 E63 E55 E55 EA5 E7E
EE54 - E60 EA5 E7F EC9 E16 E10 E7 E2
EE5E - E0 E20 E6C EFA EA5 E7F EC9 E17
EE68 - EA2 E1D EAO EE7 E23 E6C EFA EAO
EE72 - E75 EC9 E70 EDO E1C EA9 E20 EAO
EE7C - E75 EA2 E28 E66 E75 EA5 E75 EC9
EE86 - E2 E66 E76 ECA EDO EF3 E66 E7F
EE90 - EC6 E7E E60 EB1 E75 EC9 E20 EDO
EE9A - E70 EAO E0 E91 E75 E60 EC9 E74
EEA4 - EA9 E77 E91 E75 E66 E7E E60 EC9
EEAE - E7 EA9 E20 E91 E75 E66 E7E E60
EEBB - EFO EF5 EC9 E61 E30 E12 EC9 E65
EEC2 - EA9 E1 E18 E85 E6B E20 E80 E11
EECC - E11 E18 E90 E55 EC9 E65 E30 E12
EED6 - E10 EE EA9 E5 E18 E85 E6A E20
EEEO - E20 E80 E11 E18 E90 E3F EC9 E69
EEEA - EC9 E6D E10 EE EA9 E3 E18 E85
EEF4 - E40 E11 E20 E80 E11 E18 E90 E29
EEFE - E30 E22 EC9 E74 E10 E11 EA9 E5
EEOF - E20 E80 E11 E20 E80 E11 E66 E0
EF12 - EA9 E35 E91 E75 EA9 E30 E8B E91
EF1C - E91 E75 EA9 EF E85 E7 E66 E7E
EF26 - E29 E66 E75 EA5 E75 EC9 E7F EDO
EF30 - E76 ECA EDO EF3 EA2 E3 EA9 E20
EF3A - EE6 E75 EDO E2 EE6 E76 E91 E75
    
```

```

EF44 - EDO E2 EE6 E76 E91 E75 E18 EA5
EF4E - E26 E85 E75 E90 E2 EE6 E76 ECA
EF58 - EC6 E7E EA6 E6 EC6 EE72 EC6 E71
EF62 - EF9 EC6 E1 EC6 E60 EC6 E60 EC6
EF6C - E5A EDO E4 EA9 E1 E85 E6F EA5
EF76 - E14 EDO E4 EA9 E7 E85 E6 EA2
EF80 - E0 E20 E6C EFA E60 E63 E55 E55
EF8A - E55 E55 E55 E55 E55 E55 E25 E25
EF94 - E20 E6C EFA EA5 E7B EDO E6 E20
EF9E - E18 E90 E3 E20 E30 E10 EA2 E0
EFAB - E20 E6C EFA E60 E63 E55 E55
EFB2 - E85 E79 EA9 E8B E85 E7A EA4 E5
EFBC - EC9 E61 E30 E22 EC9 E6D E10 E1E
EFC6 - E91 E79 EA4 E5 EA9 E20 E91 E79
EFD0 - EDO E2 EE6 E7A EA4 E5 EB1 E79
EFDA - E91 E79 EA4 E5 EA9 E20 E91 E79
EFE4 - EDO E2 EE6 E7A EA5 E7A EC9 EBF
EFEA - EA5 E79 EC9 EDF EAO EC4 EA9 EAB
EFF8 - EA9 E8B E85 E7D EA2 E1A EAO E0
1E002 - EC9 E61 E30 EC EC9 E6C E10 E8
1E00C - E85 E7B E20 EDO E10 E60 EA5 E7C
1E016 - E2B E85 E7C E90 E2 EE6 E7D ECA
1E020 - EA6 E5 EC6 E5E EA5 E5E EC9 EFF
1E02A - EC6 E5F ECA EDO EF3 E60 EA9 EDF
1E034 - EA9 EBF E85 E7A EAO E0 EB1 E79
1E03E - E30 E26 EC9 E6D E10 E22 EA4 E5
1E048 - EAO E0 EA9 E20 E91 E79 EC6 E79
1E052 - EC9 EFF EDO E2 EC6 E7A EAO E0
1E05C - EA4 E5 E91 E79 EAO E0 EA9 E20
1E066 - EC6 E79 EA5 E79 EC9 EFF EDO E2
1E070 - EA5 E7A EC9 E8B EDO EC2 EA5 E79
1E07A - EDO E1C EA9 EF3 E85 E7C EA9 E8B
1E084 - EA2 E1A EAO E0 EB1 E7C EC9 E60
1E08E - EC9 E6D E10 E8 EA9 E0 E85 E7B
1E098 - E10 E60 EA5 E7C E18 E69 E2B E85
1E0A2 - E2 EE6 E7D ECA EDO EDE E66 E5
1E0AC - EDO E2 EE6 E5F ECA EDO EF7 E60
1E0B6 - E55 E55 E55 E55 E55 E55 E55
1E0C0 - EAO E0 EA9 EBF E85 E83 EA9 E87
1E0CA - EB1 E82 EC9 E61 E30 E18 EC9 E6D
1E0DA - EA5 E82 E18 E69 E2B E85 E84 EA5
1E0DE - E2 E69 E0 E85 E85 EB1 E82 E91
1E0EB - E20 E91 E82 E66 E82 EA5 E82 EC9
1E0F2 - E2 EC6 E83 EA5 E82 EC9 E02 EDO
1E0FC - E83 EC9 E8B EDO EC9 EA5 E5E E18
1E106 - E85 E5E E90 E2 E66 E5F E60 E63
1E110 - EA9 EBF E85 E83 EA9 E22 E85 E82
1E11A - EC9 E61 E30 EE EC9 E6C E10 E8
1E124 - E20 E30 E13 EE6 E6F E18 E90 E8
1E12E - EA5 E82 EC9 E86 EDO EE4 E60 E63
1E138 - E55 E55 E55 E55 E55 E55 E55
1E142 - EC9 E0 EFO E2B E66 E6A EE6 E65
1E14C - EC9 E3A EDO E1E EE6 E66 EA9 E30
1E156 - EA5 E6A EDO E3A EDO E12 E6A E67
1E160 - E85 E66 EA5 E67 EC9 E3A EDO E6
1E16A - E85 E67 E64 E68 ECA EDO E7 E60
1E174 - E35 E55 E55 E55 E55 E55 E55
1E17E - E35 E55 EA5 E68 EC9 E0 EFO E1F
1E188 - EE6 E66 EA5 E66 EC9 E3A EDO E12
1E192 - EA9 E30 E85 E66 EA5 E67 EC9 E3A
1E19C - EA9 E30 E85 E67 EE6 E68 ECA EDO
    
```

```

E11A6 - E63 E55 E55 E55 E55 E55 E55
E11B0 - EA5 E65 E8D EA5 E8B EA5 E66 E8D
E11BA - EA5 E67 E8D EA3 E8B EA5 E68 E8D
E11C4 - EA9 E0 E85 E6A E85 E68 EA5 E68
E11CE - E8B E30 E31 EDO E1B EA5 E67 ECD
E11D8 - E30 E2B EDO E12 EA5 E66 ECD E8D
E11E2 - E1F EDO E9 EA5 E65 ECD E8E E8B
E11EC - EFO E14 EA5 E68 E8D E8B E8B EA5
E11F6 - E8C E8B EA5 E66 E8D E8D E8B EA5
E1200 - E8E E8B E60 E63 E55 E55 E55 E55
E120A - E55 E55 E55 E55 E55 E55 EA9 E0
E1214 - EAD E9 E3 E85 E84 EC5 E82 E10
E121E - E1C E85 E82 EA5 E84 EC5 E82 E30
E1228 - E82 E18 E69 E1C E85 E82 EE6 E83
E1232 - EA9 EA8 EA5 E82 EA9 E5 E85 E83
E123C - EC5 E82 E30 EB EA5 E82 E18 E69
E1246 - E82 E66 E83 E90 EEF EA5 E83 EC9
E1250 - E2 EC6 E83 EA5 E5E E85 E5C EA5
E125A - E5D EA9 E10 E85 E5B EA5 E83 EC9
E1264 - E10 EA6 E83 EA5 E5C E18 E69 E3
E126E - E90 E2 EE6 E5D ECA EDO E72 EA0
E1278 - E2B EC6 E5C EA5 E5C EC9 E7F E70
E1282 - E5D ECA EDO EF3 EB1 E5C EC6 E5B
E128C - EC9 E61 E30 EE7 EC9 E6D E10 E33
E1296 - E18 E69 E2B E85 E5C E90 E2 EE6
E12A0 - E71 E91 E5C E60 E63 E55 E55 E55
E12AA - E55 E55 E55 E55 E55 E55 EA9 EBF
E12BA - EA9 EBF E85 E82 EB1 E82 E91 E71
E12BE - E1B EA5 E82 E69 E2B E85 E84 EA5
E12C8 - E2 E69 E0 E85 E85 EB1 E84 EC9
E12D2 - EA EA9 E20 E91 E82 EA9 E71 E91
E12DC - E39 EC9 E74 EDO EB EA9 E20 E91
E12E6 - E7B E91 E84 E18 E90 E2A EC9 E75
E12F0 - EC9 E70 EDO EA EC6 E7E EA9 E20
E12FA - E91 E84 E90 E18 EC9 E6D E30 E10
E1304 - E10 EC EA9 E20 E91 E82 EC6 E6E
E130E - E13 E18 E90 E4 EA9 E20 E91 E82
E1318 - EA5 E82 EC9 E7F EDO E2 EC6 E83
E1322 - EC9 EFA EDO E92 EA5 E83 EC9 E8B
E132C - E60 E63 E55 E55 EA2 EB9 EAO EFA
E1336 - EFA EA9 E64 E85 E82 EA2 E7F EA0
E1340 - E73 EA9 E79 E91 E73 EAO E1 EA9
E134A - E73 EC8 EA9 E7B E91 E73 EA5 E73
E1354 - E2B E85 E73 EA9 E60 E91 E73 E8B
E135E - E91 E73 E8B EA9 E7C E91 E73 E6A
E1368 - E73 E18 E69 E2B E85 E73 ECA EDO
E1372 - E82 EDO EC6 EA9 E20 EC6 E73 EAO
E137C - E73 EC8 E91 E73 EC8 E91 E73 EA5
E1386 - E29 E2B E85 E73 EA9 E20 E91 E73
E1390 - E73 E8B E91 E73 EA9 EC7 E85 E73
E139A - E91 E73 EC8 EA9 E6E E91 E73 EC8
E13AA - E91 E73 E6B E73 E60 E63 E55 E55
E13AE - E55 E55 EA5 E0 EC9 E1 EFO E2
E13BB - EDO EA EA9 E0 EA2 E25 E9D ED
E13CC - EDO EFA EA9 E66 E85 E82 EA9 E8B
E13CD - EAO E0 EB1 E82 EC9 E73 EDO E14
E13DE - E91 E82 EC6 E82 E8B EB1 E82 EC8
E13E0 - E8B EA9 E20 E91 E82 E1B E90 E8
E13EA - EA5 E82 EC9 E2D EDO EDC EAD E66
E13FA - E73 E82 EA EA9 E20 E8D E66 E8B
E13FE - E8B EC6 E0 E60 E63 E55
    
```

## Scaled a new PEEK in microcode

```

1B,A9,A9,85,00,A9,BB,85,01
100 DATA A2,26,84,02,A0,
01,B1,00,A0,00,91,00,20,32,04
110 DATA CA,D0,F2,20,32,04,
20,32,04,A4,02,88,D0,E3,68
120 DATA A8,68,AA,68,60,
18,A5,00,69,01,85,00,A5,01,69
130 DATA 00,85,01,60

```

(NB: if the ninth byte (A9) is replaced by AA and POKE 49120 used before the CALL#400 then the colours will remain as set. Also the seventh byte (1B) is the number of lines, from the top, to be scrolled so using 10 will scroll 16 lines. Note that altering these values will change the checksum — KG).

Andrew Cain,  
Cheadle,  
Cheshire.

If you've got something to crow about . . . a bit of magic that'll make the world a better place for micro users, then send it to PCN Microwaves—our regular readers' hints and tips page. We'll pay you £5 if we print it. We'll pay you even more if your little gem gets our vote as microwave of the month. Think on . . . and write to Microwaves, PCN, 62 Oxford Street, London W1A 2HG.

## Scrolling the Oric's screen

I have come up with this machine code program to use on the Oric-1. It will scroll the screen across from right to left one character everytime it is called. It leaves the leftmost column alone so you can insert your own colour codes there. It would be useful for such a game as Scramble or any other game (or even a word processor) that needs text scrolling past.

The subroutine works by first PEEKING an address in the text screen and then POKEING it into the next address. Locations #00 and #01 hold the current screen area being POKED. Index X holds the current loop variable for the lines. Before the main loop is entered Index X is saved in #02 because it has to be used as a modifier for the base address being POKED (#00,#01) to find the PEEKING address. At the end of the cycle the index Y is reloaded with the current loop variable.

The machine registers are stored onto the stack at the beginning of the routine and then recovered at the end. The subroutine at #423-#43F is used to increment the base address (#00,#01). To run the scroll routine CALL#400. To enter the routine into memory, use either a machine code monitor or a loader program like that given below.

```

10 FOR T=#400 TO #43F
20 READ A$
30 A=VAL("#"+A$)
40 CS=CS+A
50 POKE T,VAL("#"+A$)
60 EXT T
70 IF CS<>5953 THEN PRINT
  "SOMETHING WRONG SOME-
  WHERE"
80 END
90 DATA 48,8A,48,98,48,A0,

```

## 3D sine wave on Oric or Atmos

Running this program on an Oric 1 or Atmos produces a sine wave with a three dimensional appearance.

```

10 HIRES
20 FOR A=0 TO 5*PI STEP 0.1
30 B=B+1
40 CURSET B,0,1
50 DRAW 20, INT
  (SIN(A)*100+100), 1
60 NEXT A

```

Changing the multiple of PI in line 20 alters the number of waves, and changing the increment results in different densities.

Another number at the start of line 50 changes the angle from which you view the waves. The two numbers after the SIN(A) can be changed under certain conditions.

If the second number is greater than or equal to the first one the waves will be longer. If it is less, an illegal quantity error message appears.

David Webb,  
Putnoe, Bedford.

2/6/84

## The Oric turns on to disco rhythms

Here is a sound to light routine for the 48K Oric 1. Enter the routine exactly as shown, then connect a cassette recorder (or any source of music) to the Oric using the program loading lead.

With the plug half-way into the cassette you will be able to hear the music and see the lights. Run the program and switch the tape on and then adjust the volume on the cassette recorder until the lights flash in time to the music.

J Mughal,  
Southall, Middx.

```

10 P=781 : POKE#26A,10 : CLS :
  FLAG=TRUE : POKE 48035,0
20 IF PEEK(P)=2 THEN FLAG=TRUE :
  PAPER 0 : GOTO 20
30 IF FLAG THEN PAPER
  INT(RND(1)*6+1) : FLAG=FALSE
40 GOTO 20

```

## Pulling up the shades on the Oric

Colourful listings using the Oric and the Tandy CGP-115 printer are possible using the program below.

First, add the coding from line 1000 onwards to the end of your program and run this part of the program (renumber if needed).

You will be asked for a line number and two ASCII codes. If you enter 999,1,1 the program will list itself and end; if you enter a non-existent line you will be told.

To change colour on the printer enter 29,32 or 29,29 for the ASCII codes— to change colour once or twice. Note that 29 is the control code for protecting the lefthand screen columns so used once it will toggle the column protect on or off.

To change screen colour enter '27,A' where 27 is the escape code and A is an attribute code— see Appendices C and D in the Oric 1 manual. For example, for red writing use 27,65 (ASCII code for A); for magenta background use 27,85 (for something different try 27,13).

Note that the codes should only be used in a 'REM' statement, unless you want a syntax error when running. Also, the 'colour' will be removed if the line is amended, but otherwise it may be saved and loaded.

S Lowe, Stourbridge, W Midlands.

```

10 REM-----COLIST-----4/3/84-----
20 REM--S.LOWE-----
30 REM---PROGRAM FOR CHANGING COLOUR IN
  (L)LISTING----
1000 REM---CLEVER BIT---
1010 CLS
1020 INPUT "LINE NUMBER & ASCII CODES";L
  ,A,A1
1030 IF L=999 THEN LIST
1040 GOSUB 1100
1050 GOTO 1020
1100 REM---SUBROUTINE---
1110 F=1281
1120 REPEAT
1130 F=DEEK(F):B=DEEK(F+2)
1140 UNTIL B>=L
1150 IF B>L THEN PRINT "NO LINE NUMBER "
  ;L:GOTO 1170
1160 DOKE F+5,A+256*A1
1170 RETURN

```

2/6/84

- in CHR\$(130) — Green Foreground (Text & Graphics)
- is CHR\$(131) — Yellow Foreground (Text & Graphics)
- ly CHR\$(132) — Blue Foreground (Text & Graphics)
- ly CHR\$(133) — Magenta Foreground (Text & Graphics)
- o CHR\$(134) — Cyan Foreground (Text & Graphics)
- or CHR\$(135) — White Foreground (Text & Graphics)
- ly CHR\$(136) — Black Foreground (Text & Graphics)
- vn CHR\$(137) — Graphics
- m CHR\$(138) — Double Height Text
- re CHR\$(139) — Double Height Graphics
- of CHR\$(140) — Flashing Text
- CHR\$(141) — Flashing Graphics
- CHR\$(142) — Flashing Double Height Text
- CHR\$(143) — Double Height Flashing Graphics
- CHR\$(144) — Black Background
- CHR\$(145) — Red Background
- CHR\$(146) — Green Background
- CHR\$(147) — Yellow Background
- CHR\$(148) — Blue Background
- CHR\$(149) — Magenta Background
- CHR\$(150) — Cyan Background
- CHR\$(151) — White Background

### Attributed to the Oric

g  
d  
s  
h

I've been an Oric owner since early March. I had to learn without the proper manual so it was quite a surprise to find (PCN, 29 April-6 May) that you could use CHR\$(27) followed by certain letters to obtain single lines of coloured text.

I've always used the following list. For example, PRINT CHR\$(128); CHR\$(150); "HELLO" produces a black 'hello' on a cyan background.

I discovered this list purely by experimentation — there may be others.

Note that CHR\$ 138, 139, 142 and 149 need two identical program lines to achieve the desired effect.

Frank Prior,  
Chigwell, Essex.

- CHR\$(128) — Black Foreground (Text & Graphics)
- CHR\$(129) — Red Foreground (Text & Graphics)

### A new slant on Oric script

Fancying a change of character shapes I started calculating the POKEs for an italic set. It was straightforward but laborious and repetitive so I abandoned my efforts and wrote the following routine which italicises the character set automatically.

```

10 DEF FNA(X) = INT (X/2 - ((3 AND X) = 1))
20 FOR X = 46344 TO 47080
STEP 8
30 POKE X, FNA (FNA (PEEK (X)))
40 POKE X+1, FNA (FNA (PEEK X+1))
50 POKE X+2, FNA (PEEK (X+2))
60 POKE X+3, FNA (PEEK (X+3))
70 POKE X+4, FNA (PEEK (X+4))
80 NEXT

```

Alan Northcott,  
Wokingham, Berks

### A colourful clearout

The Oric 1 uses PLOT to position messages anywhere on

the screen. This just prints the characters in the message and therefore does not clear the rest of the line. SPC(n) can't be used in this way.

One neat dodge is to use an attribute that sets the foreground colour the same as the background colour. This hides any text remaining on that line. Try the accompanying program.

M Graven,  
Sale, Cheshire

4/8/83

### ROWAVES

omputing? If printed your tip will earn you a fiver.

```

10 PAPER 3: INK 2 : CLS
20 PLOT 2, 10, "THIS IS A TEST LINE"
30 WAIT 100
40 PLOT 2, 10, "NEXT LINE": WAIT 100
50 PLOT 2, 10, "LAST"+CHR$(3): WAIT 100
60 PLOT 2, 10, 3
70 PLOT 2, 20, "THE COW JUMPED OVER THE MOON"
80 WAIT 100: PLOT 21, 20, 3
90 WAIT 100: PLOT 16, 20, 3
100 WAIT 100: PLOT 9, 20, 3

```

Oric blackout — see A colourful clearout.

### Print format is POSSible

TAB on the Oric can be a problem, even when you've discovered that you need to add 10 to it to persuade it to work.

TAB can work just like the SPC function, so if you TAB(20) and TAB(30) on the

same PRINT line, the cursor is moved ten spaces to the right and not just to column 30.

One way round this is to use POS. You can subtract the position of the cursor from the column you want to tab to. For example, to put two messages at columns 20 and 30 you could use PRINT SPC(20-POS(0)); "HELLO"; SPC(30-POS(0)); "GOODBYE"

Chris Thompson,  
Orpington, Kent.

### New noise on Oric

If you're bored with the Oric's predefined sounds and find inventing your own a bit fiddly, these calls may come in useful. CALL#FB03 for a low click CALL#FB12 for a steady 'white noise'. CALL#FB30 produces a buzzing sound and CALL#FB40 gives a high pitched sound. Finally CALL#FBCO produces various sounds depending on when it is called.

S Hammet,  
Bexley, Kent

(21)

### Line copy on the Oric

If you are editing lines on your Oric, you may find this a useful dodge. Call the line for editing with EDIT line number. Then

(17)

## How to reset Oric without tears

I often have to reset the Oric's character set by pressing the reset button on the bottom of the machine.

Recently, though, I have found a useful call, CALL#F89B, which resets the characters without affecting the program in any way.

This could be very useful if you wished to have a reset within a program.

*D Thomas  
Mid-Glamorgan, South Wales.*

28/4/84

## Reversing Oric's redefined characters

Many programs for the Oric-1 make use of the user defined characters, but this has the disadvantage that when the program finishes the characters remain redefined. There is a method of correcting this and it requires just a single CALL. Insert CALL#F888 before each END OF STOP statement.

The routine at this address in

ROM does a number of things. First it resets the VIA and 8912 chips, enabling the keyboard and switching off the sound. Next the cursor, key-click and VDU are enabled, the INK and PAPER colours are set to black on white and the screen is reset to its original format. Then the character sets are defined. (Note that if you only want to reset the alternate character set, CALL#F7E0 will do the job.) Finally Caps is switched on.

Three other addresses to CALL that might be useful are: #F882, #C000, #003.

*C Hamilton,  
Belfast, N. Ireland*

## Make your Oric unstopable

While looking through the Oric's ROM I found the following useful CALL. It allows a Basic program to be executed from machine code. Just CALL or JMP (from M/C) #C765 and the effect is identical to entering RUN.

This may be useful for stopping 'software pirates' who stop auto running programs. Hence using:

```
POKE0,76:DOKE1,#C765:
DOKE#1B,#F426:CSAVE"
PROG",AUO,A0,EDEEK(#9C)
should make the program unstopable.
```

*James W Eibisch,  
Cheltenham, Gloucestershire*

## Oric goes through character change

Here is an interesting POKE for the Oric-1 that changes the complete character set. If POKED with 1, the A stays A, B turns into C, L turns into M etc. The location to POKE is at 759. POKE 759,32 converts the whole character set into lower case. POKE 759,0 changes it back again.

*Lars Lyer,  
Workington,  
Berks*

21/4/84

PCN APRIL 28 1984

## Our mistake — you can stop Oric's AUTO

Routine Inquiries (Issue 52) did not know how to stop the Oric-1's AUTO run. This is actually quite simple.

Locations #229 and #22A contain the address to which the computer jumps directly after a CLOAD; it is usually #EC03. If we change it to, say, #400 it is possible to use a machine code program to stop the AUTO run.

To make this routine work after CLOAD, type DOKE #229,#400. It is important to type DOKE #229,#EC03 before saving any programs, especially machine code.

Use the following loader program. This may also work on the Atmos.

*J D Woodcock, Deal, Kent*

### AUTO run breaking program:

```
10 FOR I=#400 TO #40B
20 READ D
30 POKE I,D
40 NEXT
50 DOKE #229,#400
60 DATA #48,#A9,#00,#85,#63,#6E,
#4C,#03,#EC
```

### Assembly language version:

Loc	Hex	Opcode	note
400	48	PHA	;Push Acc onto stack
401	A9 00	LDA#00	;Zero accumulator
403	85 63	STA 63	;Put acc at #63
405	68	PLA	;Pull acc off the
stack			
406	44 03 EC	JMP EC03;	Carry on as normal.

## Mix and merge your Oric colours

If you have ever wanted to mix colours on the Oric-1, try the following program:

```
5 HIRES
10 FOR A=1 TO 50:FILL1,1,21:
FILL1,1,21:NEXT A
20 CURSET 1,1,3
30 FOR R=1 TO 50:FILL1,1,20:
FILL1,1,17:NEXT F
```

Note that the last two parameters in the FILL command in line 30 plot the background colours blue(20) and red(17) one pixel deep. These colours are painted over the colours in line 10 to give a brown/crimson effect.

So far I have discovered lime, aqua and maroon by experimenting with other colours. Remember to change only the values in line 20 to achieve different effects.

*D Singh,  
Thamesmead, London SE2*

21/4/84

21/4/84

## The Oric as a time bomb

200  
4001

Although it's not in the manual, the Oric does have a clock accessible from Basic. The two-byte number at location #276 is decremented every one-hundredth of a second and can be used as a timer.

To set the clock, use DOKE #276,n and to read it use DEEK(#276). The maximum setting is 65535 so the clock runs for about ten minutes before repeating.

Try this five-second time bomb...

```
1000 ZAP
1010 DOKE #276,65535
1020 REPEAT
1030 UNTIL DEEK (#276) =
65035
1040 EXPLODE
```

The clock is interrupt-driven via the 6522 VIA chip. So it stops whenever the interrupts are disabled... that's when the screen is scrolled or the cassette interface is used. The clock is also useful for a Spectrum style Randomize. Try DOKE #FB,DEEK(#276): DOKE #FD,DEEK(#276).

*R Nicholson,  
Sheffield, Yorkshire*

+-----+-----+-----+-----+

Thus the CAR of symbol yields its value, and a symbol which has not been given any value defaults to itself (the value points to the same symbol). The CDR of a symbol holds the list of properties associated with the symbol; this list defaults to NIL and may be changed by RPLACD.

The function field of the symbol is NIL if no function is associated; it points to the list LAMBDA (or FLAMBDA, or MLAMBDA) for a user function; and it contains the start address for a machine-code function (the last two are differentiated by the most significant bit, which is set to 1 for a machine-code routine stored above 8000h and cleared to 0 for a user function).

The external name field (or Print-name) points to the character string which represents the symbol.

Numbers have the following structure:

Value	Properties	32-bit representation
-------	------------	-----------------------

The CAR of a number (its value) always points to itself.

The CDR of a number (its property) is T if the number is nonnegative, or NIL if it is negative.

### c) Virtual Stack

LISP is by essence a recursive language and this poses a problem when implementing it for the 6502 processor which only has an 8-bit stack pointer. OricLisp implements an unprecedented virtual stack which avoids having to simulate a 16-bit stack and makes it possible to use standard 6502 stack instructions JSR, RTS, PHA, PLA... The virtual stack consists of 64-byte segments, therefore 4 segments are present in the physical stack of the 6502. From time to time (when the EVAL routine is invoked, for example), the stack pointer position is checked to find if a segment boundary has been crossed. If so, this may result in saving a segment of the physical stack to the virtual stack or retrieving a segment from the virtual stack. However, the algorithm used is optimised for handling recursive binary trees, so segments seldom need to be transferred (no transfer occurs when the stack pointer oscillates between either side of a segment boundary).

### d) Garbage Collector

The first pass marks those pairs and atoms which can be accessed from the value, property and function fields of symbols -- marking does not apply to any symbol whose value points to itself and has no associated function definition -- or from the link stack. The second pass removes and compresses all unmarked symbols, numbers and pairs. The third pass unmarks objects and adjusts pointers to objects relocated due to compression. The last pass compresses the strings for all symbols which have not been removed by the garbage collector.

## E. Appendix E: Sample Program: Knight's Ride

The Knight's Ride is a classical poser which consists of going through each square of a chessboard once and only once, while following the rules for a Knight's movement. LISP is well suited for this kind of problem-solving. The program below provides a quick solution for chessboards of any size:

; a list of the 8 possible moves for the Knight, sorted using a heuristic method:  
(SETQ DIR '((-2 1)(-2 -1)(-1 -2)(1 -2)(2 -1)(2 1)(1 2)(-1 2)))

; first a short function builds the list of integers M,M-1...1  
(PUTD COUNTD '(LAMBDA (M) (COND ((ZEROP M) NIL) ((CONS M (COUNTD (- M 1)))))))

```

; inner function checks whether a square is outside the board:
TEST '(LAMBDA (X Y) (AND (< -1 X) (< X N) (< -1 Y) (< Y N))))

```

```

; as to build the list of locations which can be reached from M
PUTD MOVES '(LAMBDA (STEPS)
(COND ((NULL STEPS) NIL)
      ((TEST (+ (CAAR STEPS) (MOD (- M 1) N)) (+ (CADAR STEPS) (/ (- M 1) N))))
      (CONS (+ (+ M (CAAR STEPS)) (* N (CADAR STEPS))) (MOVES (CDR
STEPS))))))
      ((MOVES (CDR STEPS))))))

```

```

; build a table of such lists for all squares
(PUTD TABLE '(LAMBDA (M)
(COND ((ZEROP M) NIL)
      ((CONS (CONS M (MOVES DIR)) (TABLE (- M 1))))))

```

```

; now the program proper: look for a way through each and every square
(PUTD WAY '(LAMBDA (SQUARES REMAINING RIDE)
(COND ((ZEROP REMAINING) RIDE) ; found!
      ((NULL SQUARES) NIL) ; dead end...
      ((MEMBER (CAR SQUARES) RIDE) ; already been here...
        (WAY (CDR SQUARES) REMAINING RIDE))
      ((WAY (CDR (ASSOC (CAR SQUARES) TAB))
        (- REST 1)
        (CONS (CAR SQUARES) PARC))) ; let's try this way...
      ((WAY (CDR SQUARES) REST PARC)))) ; and other squares too...

```

```

; the main program to launch the whole thing, and that's it!
(PUTD RIDE '(LAMBDA (N TAB)
(SETQ TAB (TABLE (* N N))) ; calculate table only once
(WAY (COUNTD (* N N)) (* N N))) ; then start searching

```

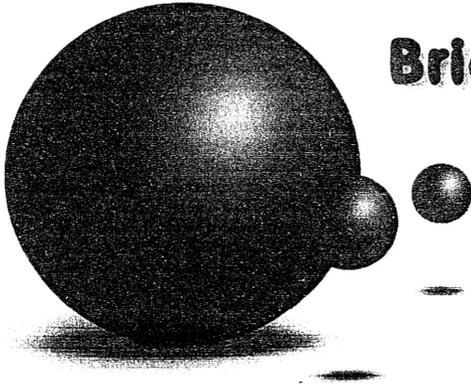
Note 1: To find all solutions, you just have to replace the first clause of the WAY function with:

```

(COND ((ZEROP REMAINING) (PRINT RIDE) NIL)

```

Note 2: A memory dump supplied with OricLisp contains an enhanced version of this program, which can be used to reach a solution more quickly.



# Brian's Poser Page/s

Time ebbs slowly away , Rhetoric soon to end.



Last Issues poser answers revealed :

- 1) Enlist , Tinsel , Listen , Silent .
- 2) I practised on a calculator , not on a telephone - check out the layout of number keys .
- 3) Atmospheric

*This issues posers .....*

- 1) The palindromic clock :

How many times a day will a 24 hour digital clock display a palindromic time ,  
Eg : 05 : 22 : 50 ?

- 2) Mental maths .

Which three digit number , when multiplied by 4 , is equal to 9 ?

- 3) What's the word ?

I'm looking for one word - forwards it means 'tense' whilst backwards it means 'Puddings' .

What's the word ?

- 4) Cryptographic or what ?

The following message uses one of the oldest known cryptographic methods known to man . Can you read what it says ?

“CYOONUGHRAAVTEUSLOALTVIEODNIST”

- 5) Common Factor ?

What have the following words got in common ?

Bow , Doe , Ruff , Threw



See you next time !