ISSUE 14 APRIL 2022 - WWW.RETROMAGAZINE.NET - FREE PUBLISHING

RetroMagazine World

future days are back

A lot of

REVIEWS

GREEN BERET

AMIGA

COLECOVISION

SPECIAL FAMILY BASIC: put the "COMPUTER" in the FAMICOM

Software: RETRO FIGHTER ENGINE for C64 - ATARI ST Basic: AES e VDI

RETRO PROGRAMMING ITALIA: How to create a BASIC game on C64 (part 2)

Columns: WAR and VIDEOGAMES - TOP10 (+ 1) intros on AMIGA

GAME BOY ADVANCE: the (S) protections - Unofficials add on for ZX Spectrum

...and then columns, interviews and much more!!!

Leave your comfort zone!

What is the comfort zone?
Cambridge Dictionary defines the comfort zone as ...*a situation in which you feel comfortable and in which your ability and determination are not being tested*...
Trying to transport this definition in the world of retrocomputing, we can immediately associate the comfort zone to machines, systems or games with which we are familiar. Probably we will immediately think of the systems that we used when we were children and / or with which we grew up and that have helped to form our experience as gamers, programmers or simple fans.

But what would happen if we decided to leave this comfort zone? What would happen if we decided to try to use and learn those unfamiliar systems? Certainly, at the beginning, we would pay the price of lack of knowledge. The commands that we are used to using almost mechanically and the operations that we usually do without even thinking about them, would probably not have the desired effect and we would feel a sense of frustration. And it is precisely at this moment, when our instinct would like us to abandon everything to return to more familiar contexts, that we must insist and move away from the established mental patterns.

Well, if we are able to do this, unexplored worlds will open up before us, where the joy of discovery could make us return to those children we were 30/40 years ago. All of us, when we picked up our first computer, felt a sort of reverential awe towards an unknown object. We had to read the manual or ask for help from more experienced friends to load a game, write a program or even just to connect cables and peripherals.

That same feeling I had when I decided to get involved with machines I didn't know at all. All my certainties vanished in an instant, just to read the contents of a disc or a cassette. Indeed, for some systems, my certainties began to waver in front of the possible configurations and options in the emulator... Fortunately, nowadays we have thousands of resources from which to draw to familiarize ourselves with these worlds unknown to us.

So what are you waiting for? Get out of your comfort zone, whatever it is, and try to use systems you don't know yet. A new world is waiting for you. A world unexplored for you, but with that richness of colors that has already fascinated thousands of users, with those games that have driven generations crazy and with those commands that have kept programmers busy for entire nights!

Leave your comfort zone and you'll see some great things!

Francesco Fiorentini

TABLE OF CONTENTS

People involved in the preparation of this issue of RetroMagazine World (in no particular order):

- Alberto Apostolo
- Dr. Andrea Q.
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Mic the Biker Novarina
- Francesco Fiorentini
- David La Monaca
- Leonardo Miliani
- Barbara “Morgana” Murgida
- Takahiro Yoshioka
- Roberto Del Mar Pirazzini
- Ingrid Poggiali
- Gianluca Girelli
- Giampaolo Moraschi
- Querino Ialongo
- Gianluca Alberico
- Felice Nardella
- Beppe Rinella
- Alessandro Albano
- Cover image: **Giuseppe Mangini**
- Cover layout: **Carlo N. Del Mar Pirazzini**

Family BASIC - Put a "computer" in the Famicom

by Takahiro Yoshioka (with Carlo Nithaiah Del Mar Pirazzini)

When Nintendo released its first console, the Famicom, the full name of the console was Nintendo Family Computer. In the early 1980s, the line between video game consoles and home computers was blurred.

Some consoles, like the Odyssey 2, had a full (membrane) keyboard, but were more like video game consoles.
Some computers like the Commodore 64 had a full keyboard and a disk drive, but they could be loaded with games.

At the launch of the Famicom in 1983, the only software available for the machine were arcade game ports.
By the following year (1984) Nintendo had released a new Family Computer product that was intended to do more than just play another video game.
This product was released as Family BASIC on July 21, 1984, but was only released in Japan.

The Family BASIC package included two hardware components. The first was the Family BASIC cartridge, the second was the Family Keyboard.

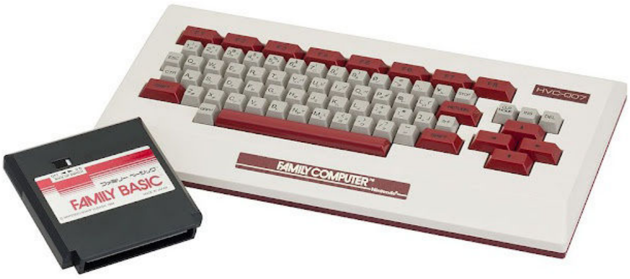


Fig. 1 - Picture from Wikipedia

The Family BASIC cartridge was unique in its time. It contained 32 KB of program rom (PRG) and 8KB of character rom (CHR-). In a second version it was provided with 2KB or 4KB of working Ram (W).

There was an additional logic chip inside to disable the PRG-ROM chip when reading or writing to W-RAM since both memories were located on the CPU address and data buses.

Two other features of the cartridge were a software backup on-off switch in the battery. When the switch was on, the contents in W-RAM were not lost when the Famicom was turned off.

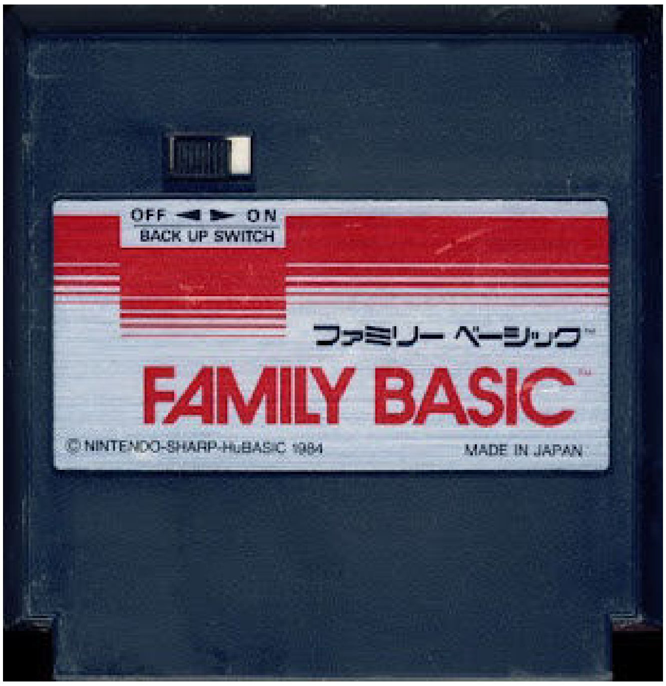


Fig. 2 - Family Basic - Picture from Bootgod

The second unique feature is that the internal battery was totally user-replaceable because it was AA size.
There were four versions of the Family BASIC software released. The first three, v.1.0, v.2.0A, and v.2.1A, looked the same externally, contained 2KB of W-RAM, and were also sold with the Family Keyboard. There was also a version for Sharp My Computer C-1 Tv called Playbox BASIC.

The final v.3.0 version was released in 1985 with 4KB of W-RAM and sold separately from the Family BASIC package and keyboard.

The keyboard had 72 keys and was a mechanical structure but with the size of the buttons much smaller than the classic keyboards. It was similar in appearance to that of the C64 and the Sharp X-1 and connected to the Famicom through the 15-pin expansion port.
It was designed to have a matrix of two "columns" and eight "rows". Each key was unique and had individual functions.

It was also equipped with a pair of 3.5 mm mono audio

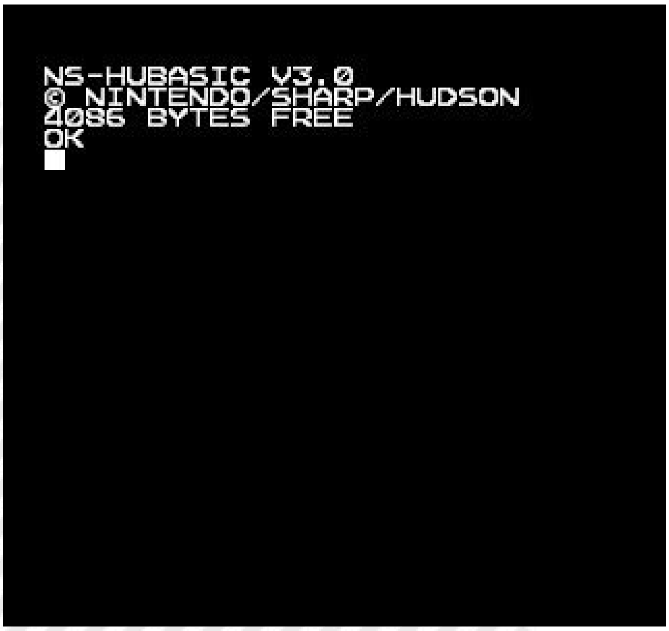
Fig. 3 - Family Basic V3 - Picture from Bootgod

jacks labeled "write" and "red" in the early versions and later as "Save" and "Load" in the post-1984 versions. To these jacks it was possible to attach a data recorder, purchased separately, which allowed you to record the work done.

The BASIC

The BASIC family was a port of Hudson Soft's HU-BASIC, cross-assembled on Sharp's line of computers. HU-BASIC, like most home computer BASICs, was based on Microsoft BASIC.

Many familiar commands and conventions for users familiar with a dialect of BASIC descended from Microsoft. The standard HELLO WORLD program is entered and executed in the usual manner. LIST, NEW, CLS, SAVE, and LOAD work as expected. (You can use Shift + Clr/Home to clear the screen).



```
NS-HUBASIC V3.0
© NINTENDO/SHARP/HUDSON
4086 BYTES FREE
OK
█
```

Fig. 4 - The Basic interpreter

There are singular commands and differences in this dialect of BASIC. Unlike a Microsoft BASIC, Family BASIC only works with integers, not decimals or floating point. If you type STAMP 3/2 and press Enter, the result is 1, not 1.5. Lowercase English text is not present, but the kana key will allow you to type katakana or English character by character.

There is a SPRITE command to define and display sprites, a MOVE command to move sprites, a PLAY command to play music, a PALET command to change sprite palettes or background/text, and STRIG and STICK commands to take input from the gamepad's action and directional sprites.

There are two important commands LOAD and SAVE in Family BASIC. The normal LOAD and SAVE commands load a program from tape or save a program to tape. Background graphics are loaded into the internal 2KiB of video RAM inside the Famicom. Use the LOADS and SAVES commands to load or save background graphic data to/from tape for Family BASIC v3.

The Family BASIC package included an extensive manual to help teach people how to program in BASIC. It also had a sheet showing all the graphics available for backgrounds and sprites. Of course, the manuals are in Japanese as is much of the text displayed in the program itself.

You can find scans of the manuals and several translations at this web address: https://www.famicomworld.com/forum/index.php?topic=11959.0

FAMILY BASIC v.1.0 - v.2.1

A very "time friendly" system. At power up, a "Computer" screen appears, an "Operator" screen, and F1 to F4 at the bottom (Fig. 5).

By pressing any key, the computer will begin to explain to the user what it is capable of doing. After three screens of text you will be asked to register your name.

After a text screen where the next menu is explained you will be asked if you want to start "GAME BASIC". Press F1 (Yes) and then enter.

This will take you to a menu where you can choose to access either the BASIC or the BG Graphic program. If you hold down the T key when starting the Famicom, you

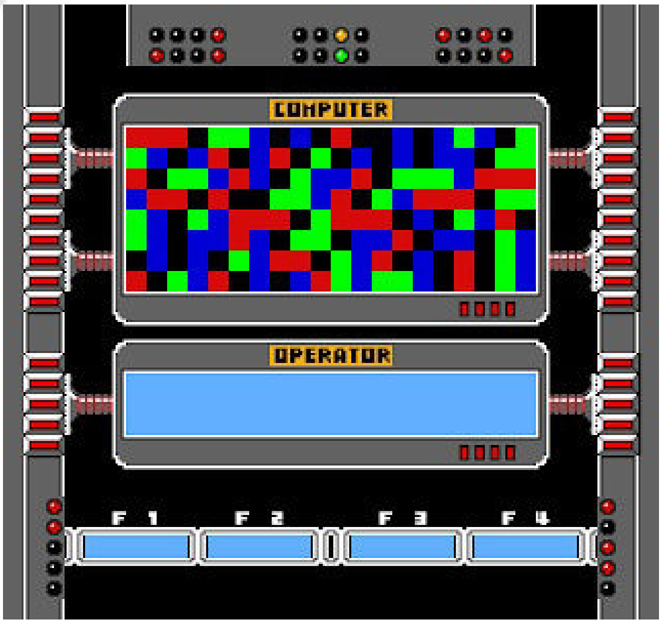
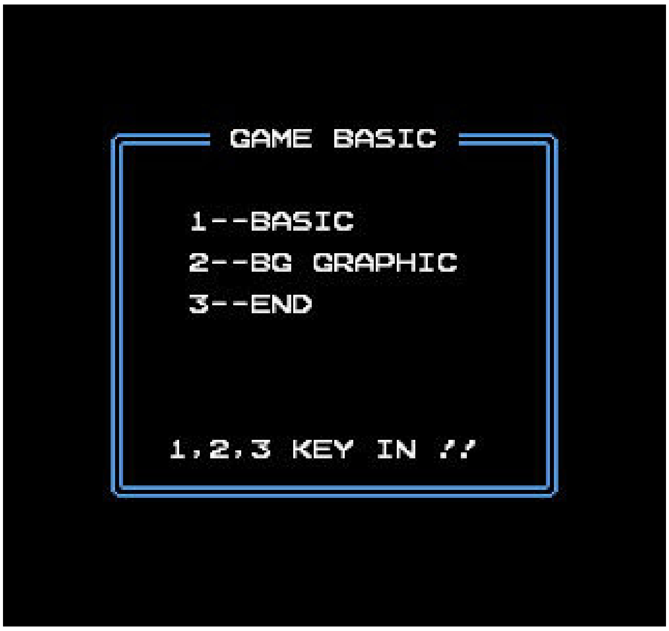


Fig. 5 - Operator and Computer screen

can skip the home screen and go directly to the BASIC/ BG menu (Fig.6).

The button immediately below Return is the Kana button. This switches the keyboard between alphanumeric mode and Katakana mode. In this mode, only Katakana is available from the alphanumeric keys. If you want numbers or symbols or punctuation marks (except those used in Japanese), you must switch back to alphanumeric mode. The GRPH key will give you the version of the character. The Shift key is used to share kana with the YUIOP keys, giving you the version of those characters and the smaller version of the character if shown on the keyboard. If the cursor is a blinking box, the keyboard is in Katakana mode. If it is a blinking square, then it is in alphanumeric mode.

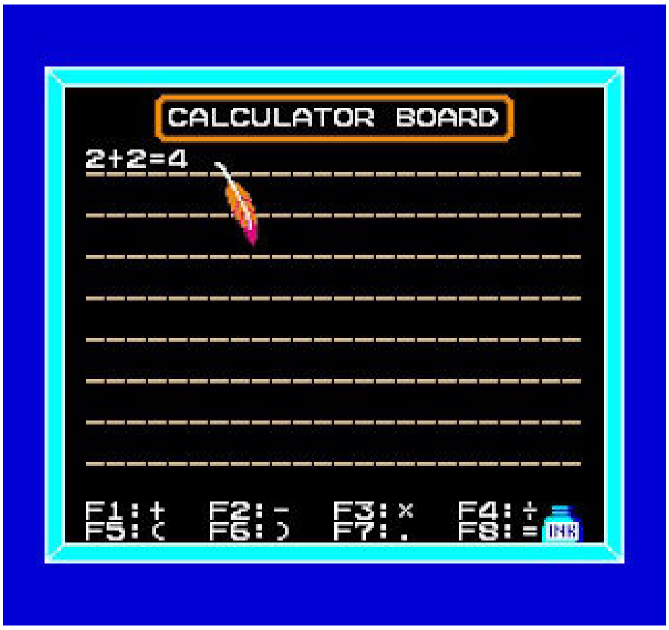


```
GAME BASIC

1--BASIC
2--BG GRAPHIC
3--END

1,2,3 KEY IN !!
```

Fig. 6 - The Basic/BG Grpahic menu

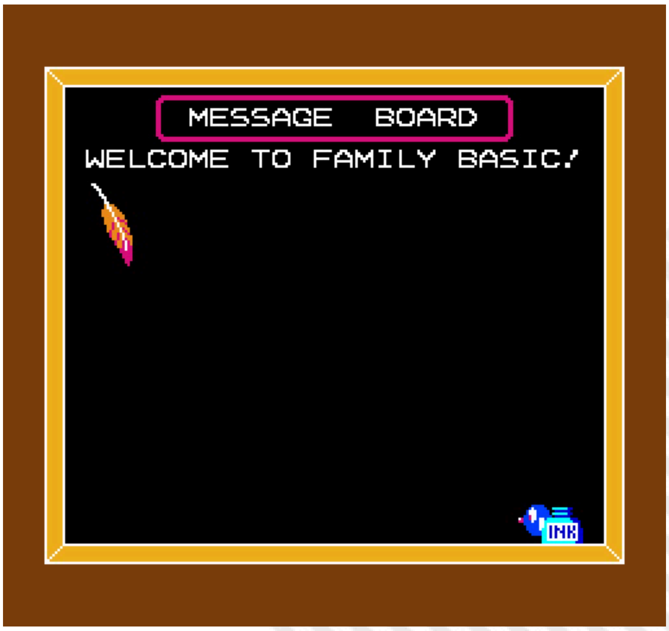
The BG Graphic program allows you to use the navigation keys to display panes on the screen. Using the up/down/ left/right arrow keys we can place the cursor on the frame we want to edit and press the space bar to set the frame.

Family BASIC also contains three other programs, the Calculator, the music board, and the message board. The calculator allows us to solve equations more easily from the BASIC prompt, the music board allows us to compose music, and the message board allows us to add a variety of annotations. Data from these programs can be saved to W-RAM but not to tape.



```
CALCULATOR BOARD
2+2=4

F1:+   F2:-   F3:×   F4:÷
F5:(   F6:)   F7:.   F8:=
```

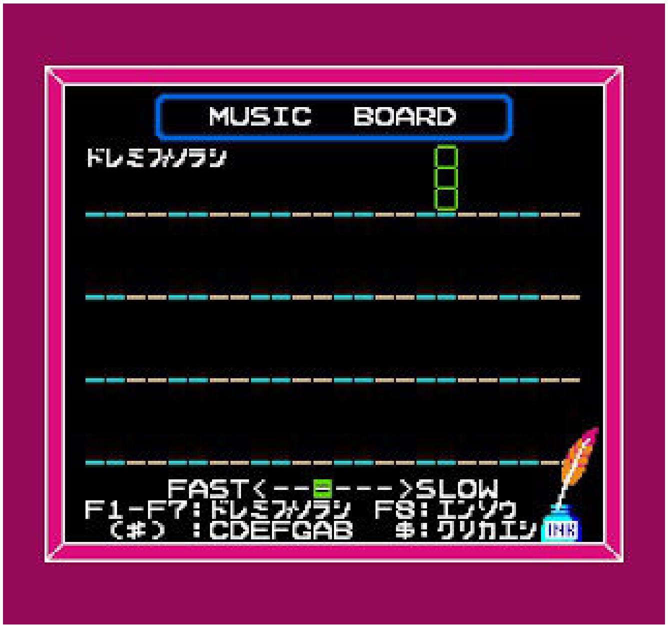
Fig. 7 - The Calculator program



```
MESSAGE  BOARD
WELCOME TO FAMILY BASIC!
```

Fig. 8 - The Message Board program

To access these programs you need to press the F2 key at startup and select which application you want to use.



MUSIC BOARD

ドレミファソラシ

FAST<--■--->SLOW

F1-F7:ドレミファソラシ F8:エンソウ

(#) :CDEFGAB $:クリカエシ

Fig. 9 - The program Music Board

Family BASIC V3

This version improves the application's internal navigation and lets you access each program more quickly.

There are also 4 built-in mini-games that can be modified and they are: Heart, Penpen Maze, Mario World and Star Killer. Several options have been added in BGTOOL and the graphics tool. A Crash command has been added to detect collisions between objects.

The legacy of Family BASIC

In Japan, the cost of a "real" computer was very high, even for 8-bit machines like Sharp's X-1 or NEC's PC-8801 or Fujitsu's FM-7. Most Japanese families were not ready for a plunge of thousands of dollars to bring the home office with them. Game consoles were much, but much cheaper devices and could work better than most home computers. Family BASIC was more expensive than a standard video game cartridge (¥14,800, same price as the Famicom) but less than what the "professional" market presented. Family BASIC was a tool to help children learn while having fun.

It was one of the few pedagogical programs published by Nintendo, along with Donkey Kong Jr. Math and Popeye's English Lesson, during the early years of the Famicom. In the manuals, Nintendo had to document how the Famicom worked to a greater extent than it otherwise would have had to explain. This helped some programmers like Satoshi Tajiri, the creator of Pokemon, get started by allowing him to explore the hardware and use the concepts he learned to help program his first game, Quinty.

Family BASIC, although a modest success, did not lead to a spring of interest in programming in the Famicom. The Famicom, despite its name, was a video game console, not a home computer. Cassette storage was something the U.S. had left behind in the 1970s with little nostalgia. While Family BASIC peaked at 4 KiB in 1985, by 1977 the minimum configurations of the Apple II, Commodore PET and TRS-80 were as low as 4 KB.

In order to compete with the market, the Family Computer needed a disk drive and, despite the creation of the Famicom Disk System, no programming software was ever released. When Nintendo was planning the release of the Famicom in the United States, they thought of

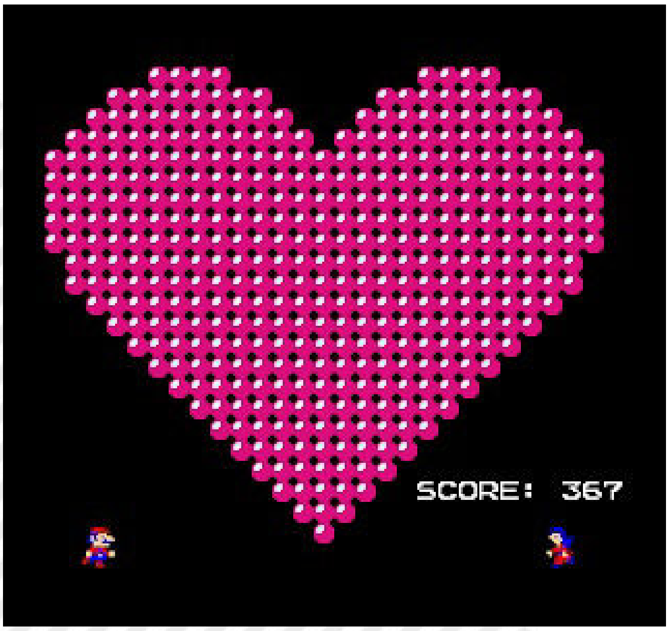


Fig. 10 - Heart

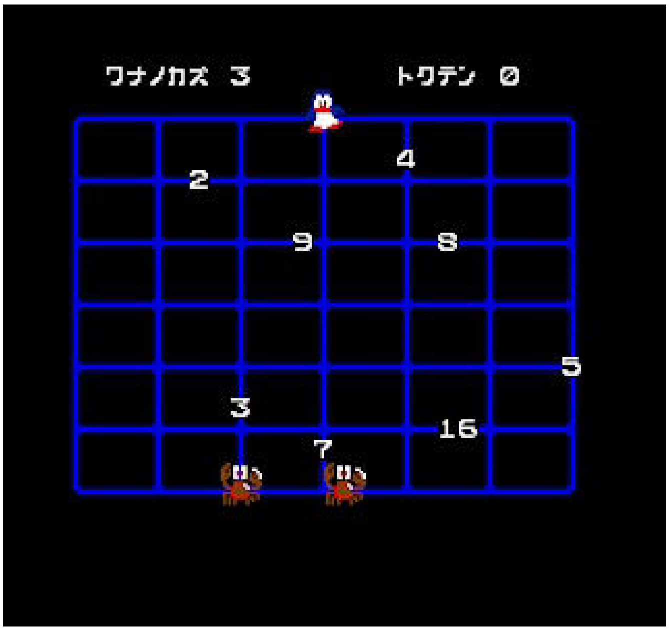


Fig. 11 - Penpen Maze

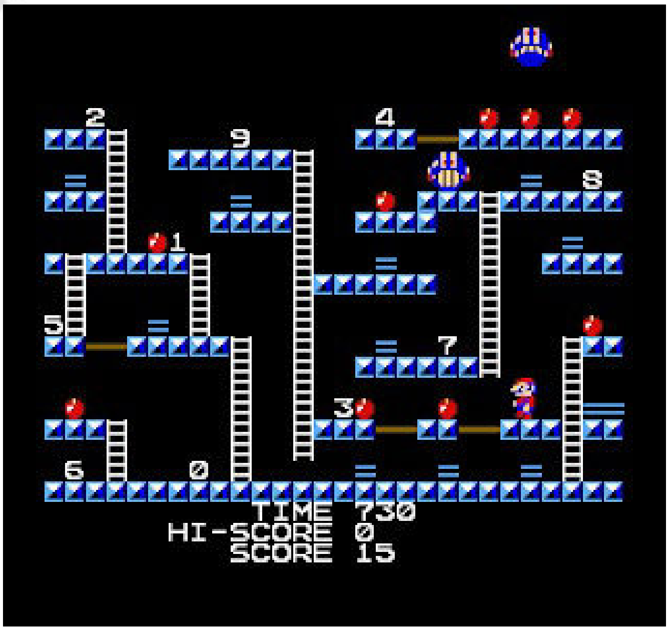


Fig. 12 - Mario World

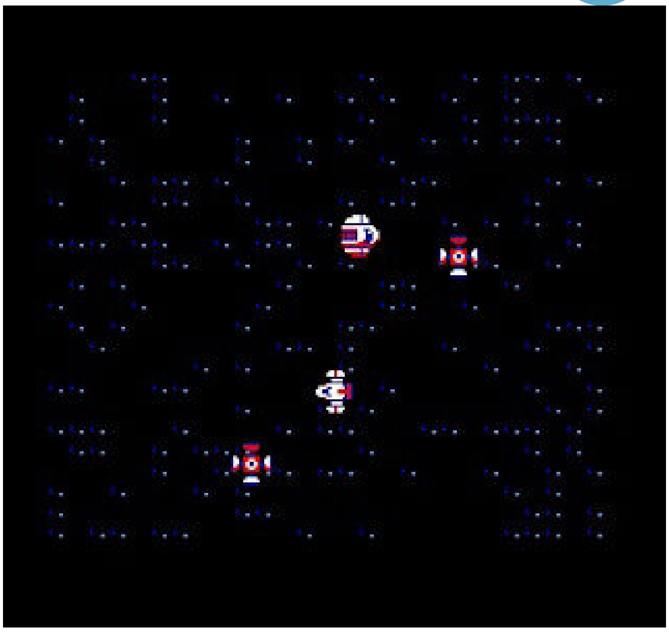


Fig. 13 - Star Killer

adding the Family Basic into the project, calling it the Advanced Video Systemha. The prototype also featured a keyboard and a data cassette recorder. The keyboard prototype has 72 keys with the katakana symbols removed and the kana key turned into a Caps Lock key. Presumably if the Family BASIC software had been translated into English, it would have had the ability to display lowercase English text. It was not released and nothing was released for the NES officially or unofficially during its lifetime.

If Nintendo had released the keyboard for the NES, the keyboard would have had to work very differently. The NES has fewer input and output ports via its controller ports than the Famicom does via its expansion ports. Also, the controller ports have to share input lines with the controllers. With cheap 8-bit computers like the Commodore 64 still around in the late 1980s, it didn't make much sense to introduce a programming cartridge.

Bibliography

- Famicom World: https://famicomworld.com/
- Famibe No Yosshin (May 1993). "ZACNER II"
- Szczepaniak, John (August 2012). "A basic history of BASIC on its 50th birthday"
- Kojima, Ikuo (1994). Pseudorabies Publishing. p. 74.

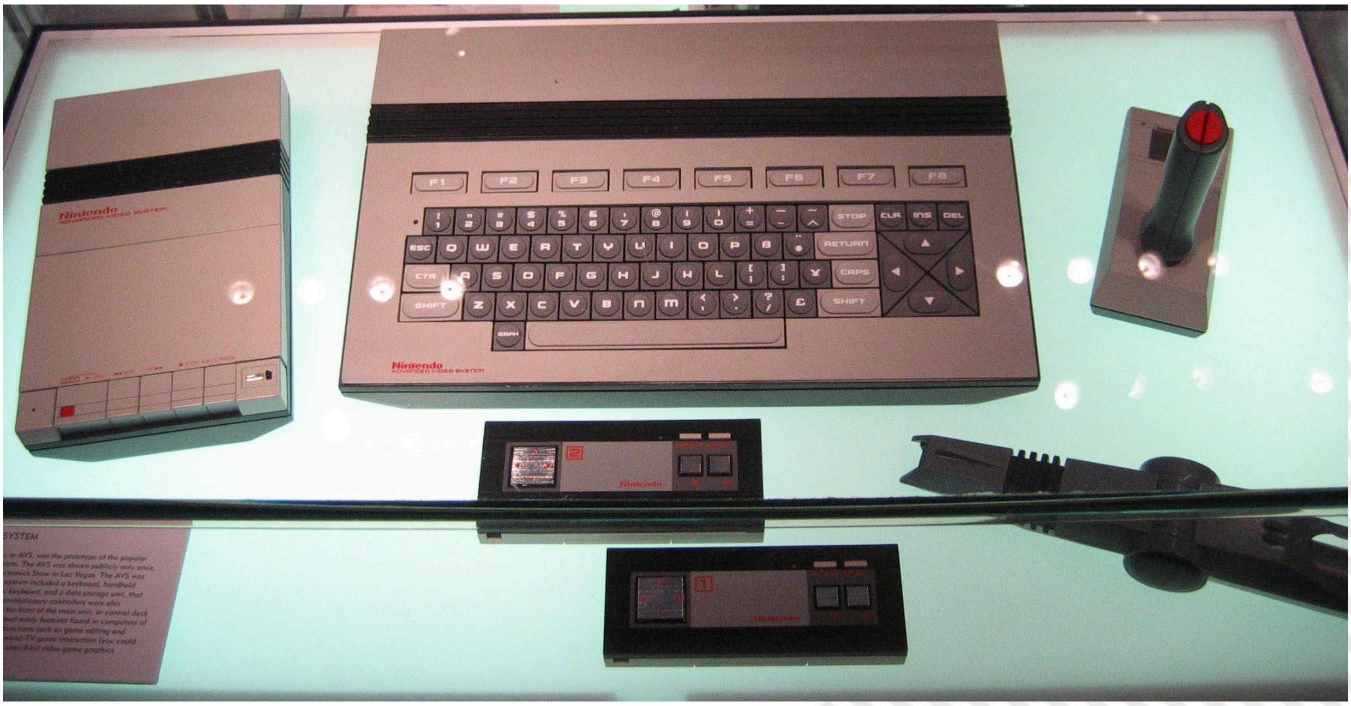


Fig. 14 - The prototype of the Advanced Video System

GameBoy Advance - the (un)protections

by Dr. Andrea Q. - www.retrofixer.it

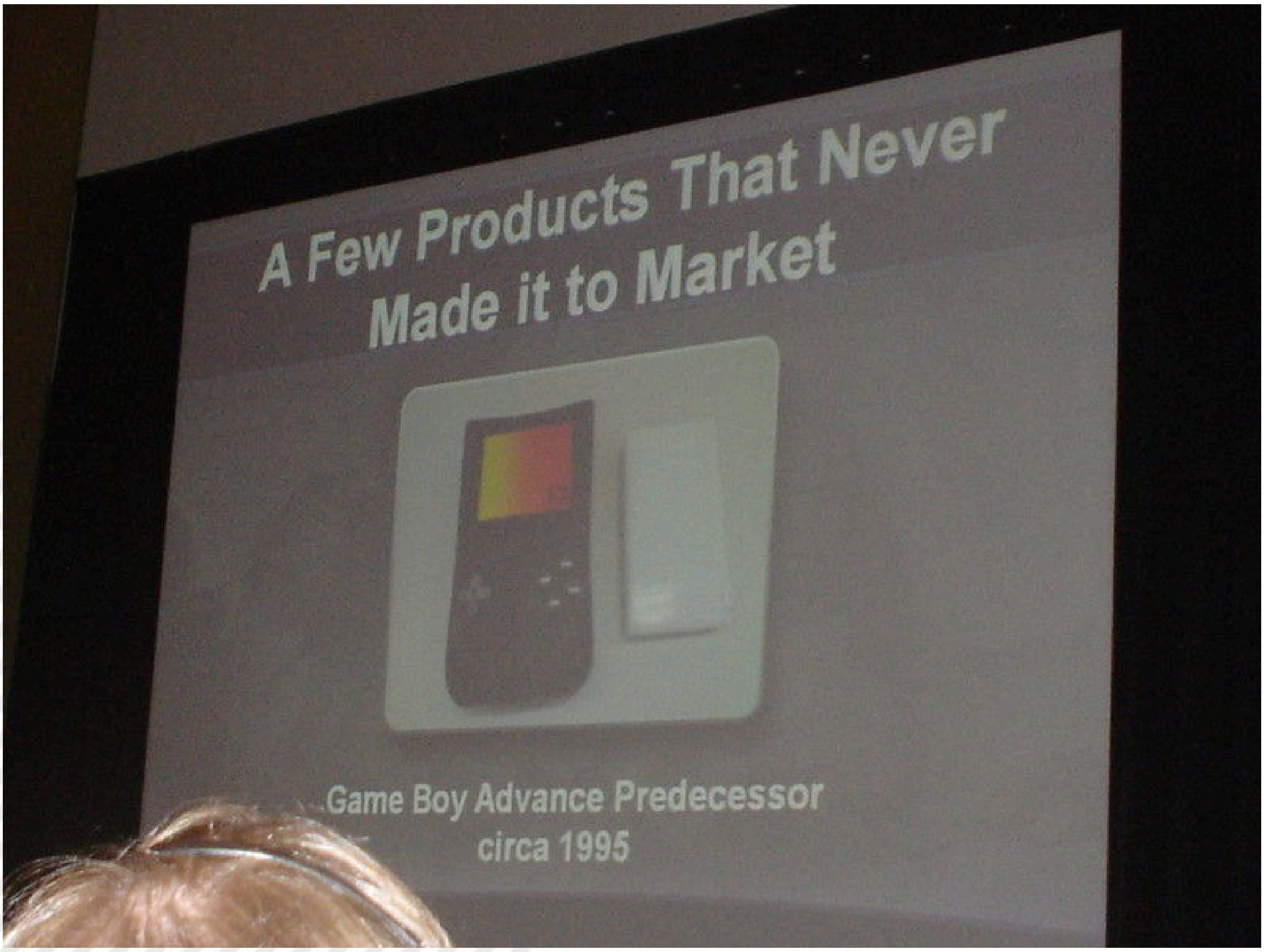
Youtube: https://www.youtube.com/channel/UCEw0CQ8LKyA9jVvWXkEwp4Q

The Game Boy Advance (GBA), product code AGB-001, saw the light for the first time in 2001 and had as its predecessor a never released commercially "Project Atlantis", perhaps successor to the Game Boy Color (in the big picture below you can see him on the left in one of his rare appearances next to a DSi).

PROTECTIONS

The console was region free and had no particular protection systems (if not the usual checking of the logo both in the internal ROM of the BIOS [offset $&3290] and in the cartridge [offset 0x0004, for 156 bytes - checked byte by byte] but it was the first, if we do not consider the additional device 64DD for N64, to have its own operating system. A sort of mini-BIOS that contained, in only 16KB, the logo and could perform quick memory copy procedures, decompressions, some basic mathematical operations, some audio functionality (probably used by only 3 games) and a handful of low-level hardware interactions.

The "CLASSIC NES SERIES" cartridges (the physical version of future "Virtual Console" titles) appear to contain


A Few Products That Never Made it to Market
Game Boy Advance Predecessor
circa 1995

particular functions that would cause compatibility issues if used with non-original Game Boy Advance hardware, specifically the functions:

- CPU pipeline (self-modifying code that shall NOT affect prefetched opcodes)
- STMDA write to I/O ports (writes in INCREASING order, not DECREASING order)
- SRAM detection (refuses to run if SRAM exists; the games do contain EEPROM)
- ROM mirrors (instead of the usual increasing numbers in unused ROM area)
- RAM mirrors (eg. main RAM accessed at 2F00000h instead of 2000000h)

As in the previous portable consoles, in the GBA the BIOS cannot be read; in fact the copy or memory access functions of the BIOS are controlled so that they cannot access the area where it is stored... All except one, the software interrupt $1F, also called MidiKey2Freq, used to convert musical notes from MIDI to a frequency audible to the human ear.

This fact did not escape the coder Dark Fader that already in 2001 exploited it to obtain, in a rather slow way and byte by byte, the complete BIOS dump on flash cartridge. This system, until recently, was the only way to make the dump until more recently, in 2016, the dev endrift managed to use a ROP technique to achieve the same goal.

What is a ROP technique?

Making things simple we can say that, with judicious programming, ROP chains can get around the fact that memory is not "executable". More in detail: recent CPUs don't allow to execute arbitrary code in memory: this means that if you want you can copy data in memory but this doesn't mean that these data can be executed; to do this you need to copy these data in specific areas of memory defined "executable" that are set as such by the Memory Management Unit MMU (hardware that can be separate or integrated in the CPU). ROP techniques bypass this "obstacle" by precisely selecting the end of several functions that manipulate system state in specific ways and chaining the end of a function call with a jump to the end of another function that performs another specific set of operations.

With its new dump method (it also dumps the BIOS directly into the SRAM of a flash cartridge) endrift shows us that a software vulnerability is not always needed to exploit (exploit) a system because hardware flaws go much deeper.

Given the lack of significant protections and the relative ease with which to replicate a GBA cartridge, clones of games for this consoles were quite popular at the time, especially in Asia.

This article was short and not very intense but the next one will compensate for that. In fact, we'll feature a discussion about Game Cube protections!

ATTENTION: LIABILITY DISCLAIMER

The information contained in this article is for informational purposes only. This documentation is not guaranteed to be error-free. If this information is used to modify your hardware, it is your responsibility to take all necessary emergency, backup, redundancy, and other measures to ensure safe use. RetroMagazine World assumes no responsibility for any damage caused by the use of the information in this article.

Unofficial add-ons for the ZX Spectrum

by Alberto Apostolo

In the 80s of the 20th century, several companies produced expansion cards that allowed you to upgrade the Sinclair ZX Spectrum, save data and programs on floppy disks, print graphics and even make the computer "talk"!
In this article I wanted to describe what caught my attention the most.

Sandy

Sandy was an Italian company based in Senago, a small town north of Milan.
In the 80s of the 20th century, it marketed a series of interfaces for 5.25-inch (100 Kb, 200 Kb, 400 Kb) and 3.5-inch (200 Kb, 400 Kb) floppy disk drives.
After connecting and switching on the computer, control switched to the "SP DOS" operating system (fig. 2). A password was required to access the contents of the diskettes.

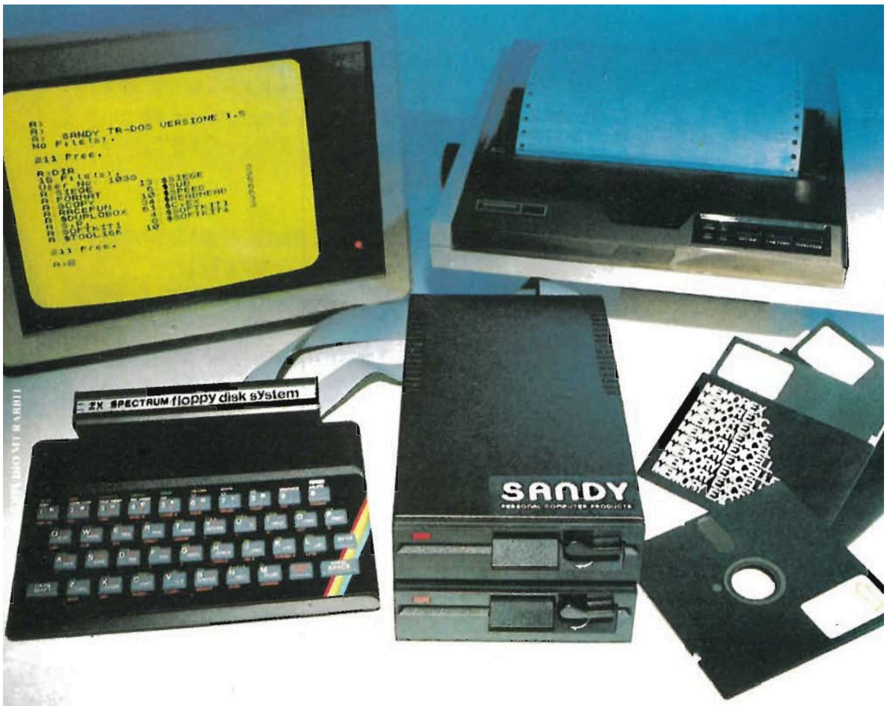
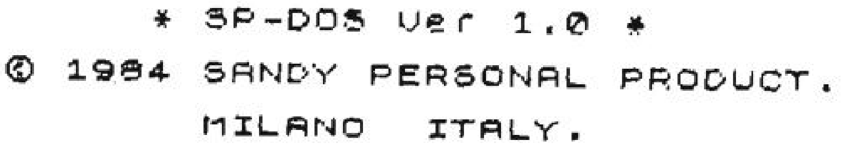
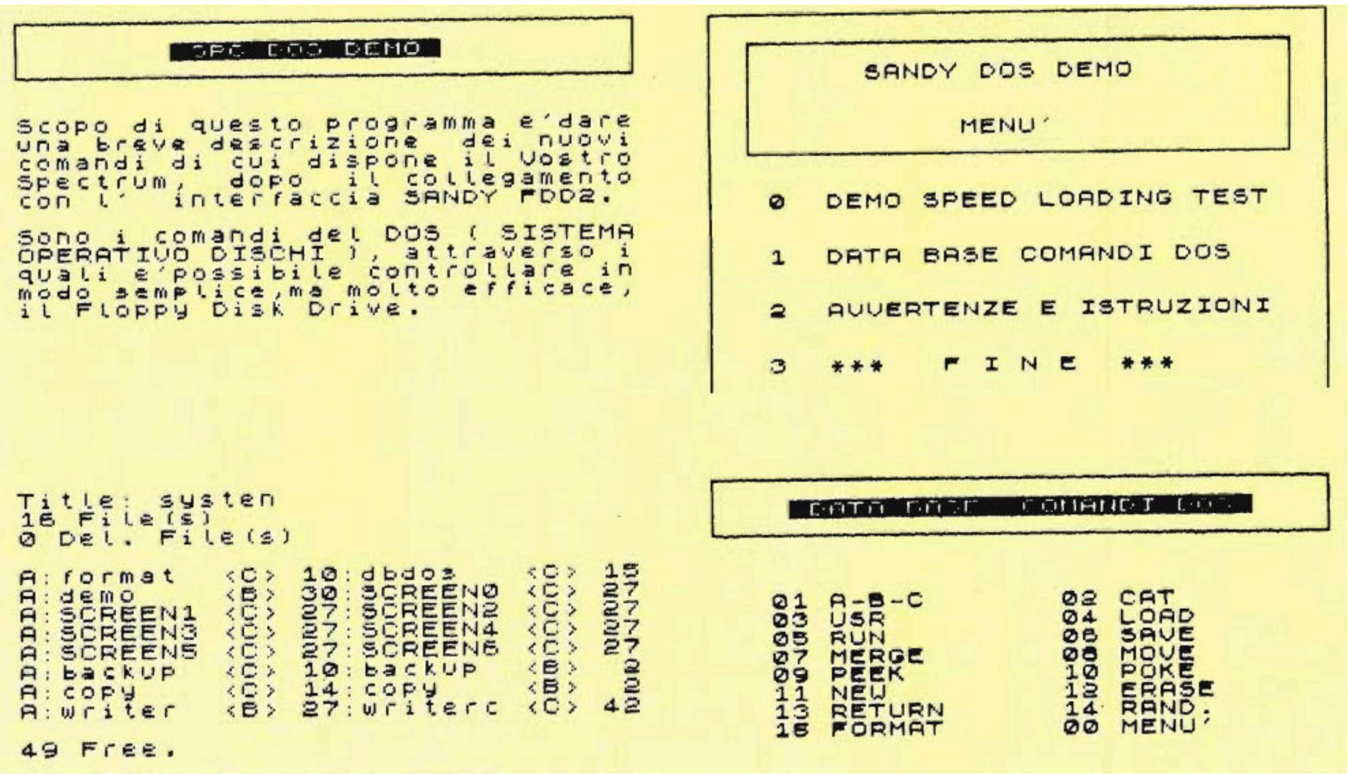

SANDY TR-DOS VERSIONE 1.5
ZX SPECTRUM floppy disk system
SANDY
PERSONAL COMPUTER PRODUCTS

Fig.1



```
* SP-DOS Ver 1.0 *
© 1984 SANDY PERSONAL PRODUCT.
MILANO  ITALY.
```

Fig.2



```
SPC DOS DEMO

Scopo di questo programma e'dare
una breve descrizione  dei nuovi
comandi di cui dispone il Vostro
Spectrum,  dopo  il collegamento
con l'  interfaccia SANDY FDD2.

Sono i comandi del DOS ( SISTEMA
OPERATIVO DISCHI ), attraverso i
quali e'possibile controllare in
modo semplice,ma molto efficace,
il Floppy Disk Drive.
```

```
Title: systen
16 File(s)
0 Del. File(s)

A:format    <C> 10:dbdos    <C> 15
A:demo      <B> 30:SCREEN0  <C> 27
A:SCREEN1   <C> 27:SCREEN2  <C> 27
A:SCREEN3   <C> 27:SCREEN4  <C> 27
A:SCREEN5   <C> 27:SCREEN6  <C> 27
A:backup    <C> 10:backup   <B>  2
A:copy      <C> 14:copy     <B>  2
A:writer    <B> 27:writerc  <C> 42

49 Free.
```

```
SANDY DOS DEMO

MENU'

0  DEMO SPEED LOADING TEST

1  DATA BASE COMANDI DOS

2  AVVERTENZE E ISTRUZIONI

3  ***   F I N E   ***
```

```
DATA BASE   COMANDI DOS

01 A-B-C          02 CAT
03 USR            04 LOAD
05 RUN            06 SAVE
07 MERGE          08 MOVE
09 PEEK           10 POKE
11 NEW            12 ERASE
13 RETURN         14 RAND.
15 FORMAT         00 MENU'
```

Fig.3

SANDY DISCO VERS.3

Floppy disk: 3.5 pollici

Capacity: 1Mb, formatted 800Kb

Tracks: 80 on 2 sides

Transf. speed: 250 Kbit/s

LOAD/SAVE speed: 15 Kbyte/s

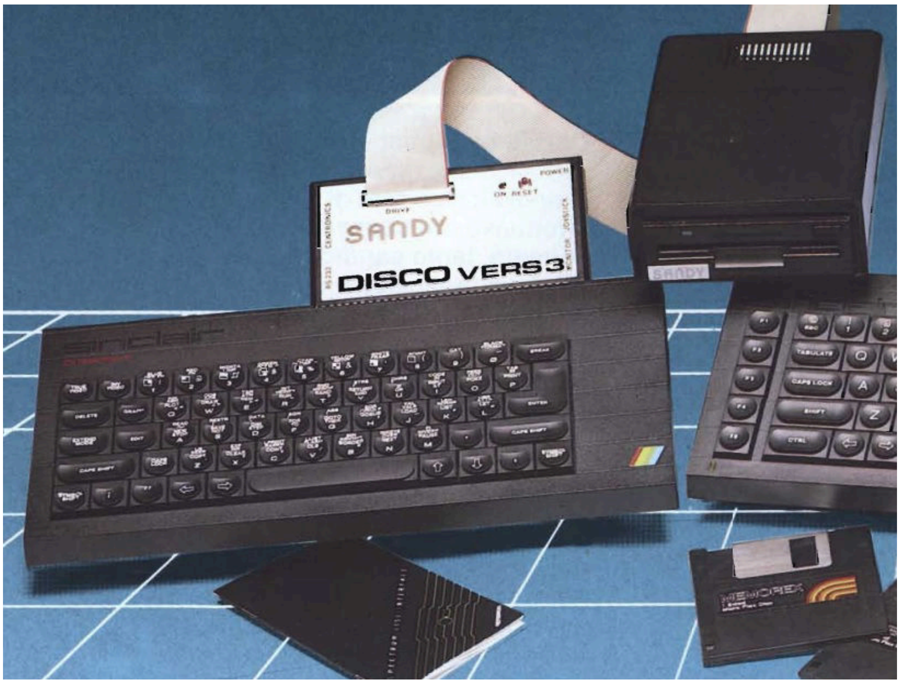


Fig.4

OPUS DISCOVERY

1 3.5 inch Floppy disk drive
(250Kb double density)
1 connector for drive #2
1 parallel interface
1 RS 232 interface
1 joystick interface
(Kempston and Atari compatible)
1 video-monitor interface
1 external connector
1 built-in power supply

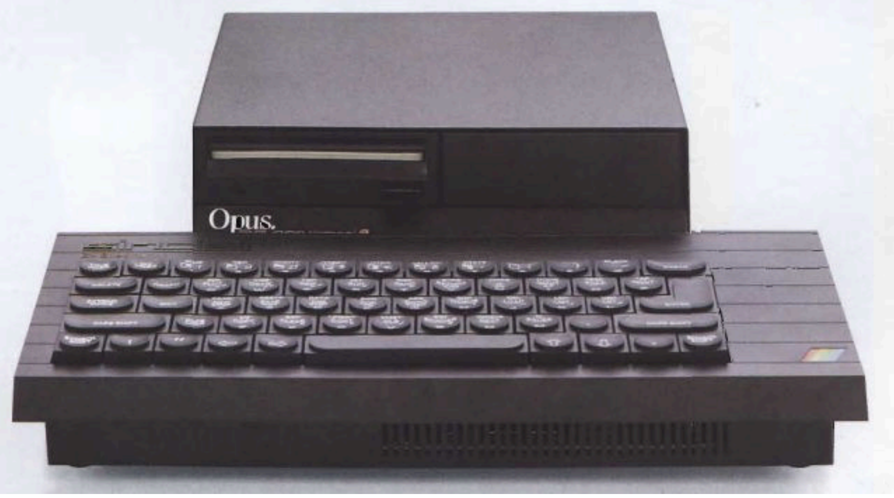


Fig.5

For the management of files on disk, the Sandy interface made available (via a "ghost" ROM) various BASIC commands. Some were derived from Sinclair BASIC (SAVE, LOAD, MERGE, CAT, FORMAT, etc.), others were added (INIT, PUT, GET, etc.). The demonstration program supplied with the interface was very useful (fig. 3).

Later on, a more advanced version with a 3.5 inch floppy disk [Spe85] was marketed (fig. 4), which had several interfaces (RS232 TTL, Centronics, Kempston), an RCA Plug monitor output and a 56 pin connector.

The floppy disk was separate and self-powered, and could be connected to cards designed for other models such as the Sinclair QL, MSX, Amstrad, etc.

For those who want to know more about it, there is an excellent review written (only in Italian) by Giancarlo Errichelli for the Italian magazine "Sperimentare con l'Elettronica e il Computer" [Err85].

Opus Discovery

In 1985, Opus Supplies Ltd (a British company based in Redhill, Surrey) marketed, for £199.95, a 3.5-inch floppy disk interface with 250Kb capacity (fig. 5).

Discovery came with utilities for format and verify operations and RAM DISC management.

In addition, it was compatible with the ZX Interface 1 MicroDrives (DOS Discovery retained the commands provided for ZX Interface 1).

Performance was lower than ZX Interface 1 (15 Kbaud vs. 19.2) but the emphasis was on the greater robustness of floppy disks compared to MicroDrives.

DISCiPLE

DISCiPLE was an interface for the Sinclair ZX Spectrum (fig. 6,7) designed by Miles Gordon Technology. From 1986 it was marketed by the British company Rockfort Products (based in London).

Outwardly resembling the ZX

Interface 1, it was proposed as a super interface with many functions.

It had a Centronics parallel port, a Shugart bus for floppy disks, two joystick connections (Kempston and Atari compatible), a 3.5 mm audio connector and a connection for ZX Interface 1.

It also had a "magic" button for getting "snapshots" of the state of the machine (memory and registers) that could be saved to file (very useful in games to restart from a previously saved situation).

The commands to manage the files (Sinclair BASIC extensions) constituted the GDOS.

Unfortunately, being very sophisticated it was also expensive (£73 + VAT, without the floppy drives). A special button could "inhibit" the board (if necessary) to allow other peripherals to function properly.

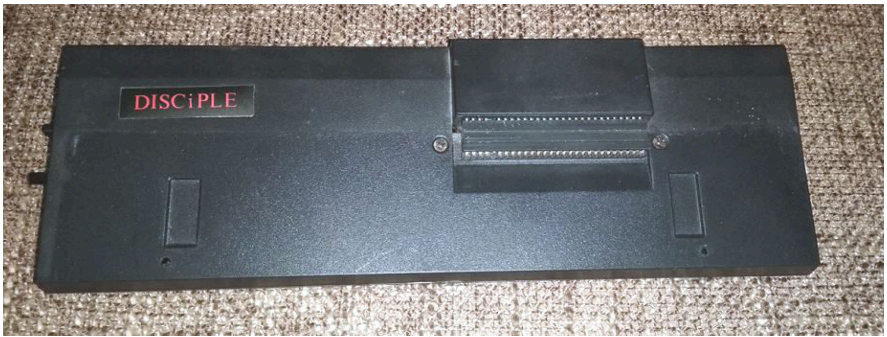


Fig.6

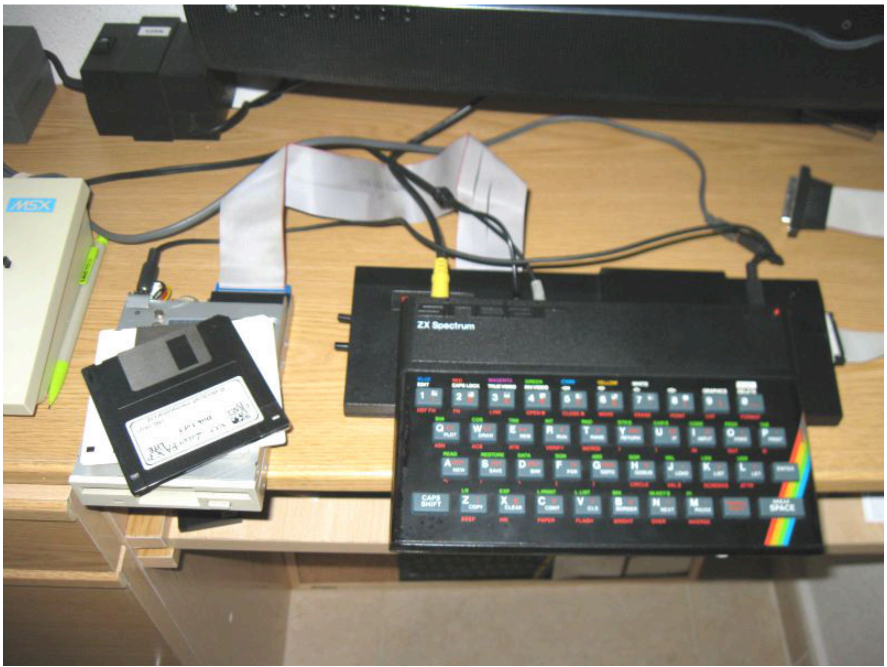


Fig.7

Cheetah Sweet Talker

Sweet Talker (fig. 8) was a voice synthesiser for connection to the Sinclair ZX Spectrum, manufactured by Cheetah Marketing LTD (a British company based in London). Its operation was based on the 64 allophones of the English language (fig. 9).

Allophones are the basic units of sound into which any word can be broken down.

By combining them appropriately, a potentially unlimited vocabulary was available.

However, such a system had two disadvantages. Firstly, the sound quality was not very good, and secondly, it was caused by the allophones of the English language (the pronunciation of words belonging to other languages had a nice British accent). Finally, it lacked a real management software and was not compatible with MicroDrives.

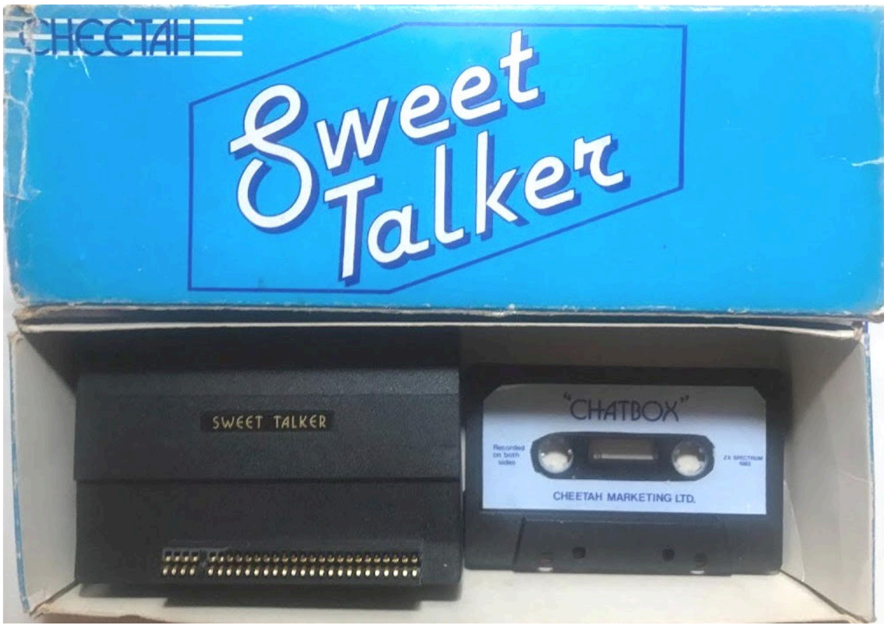


Fig.8

Seikosha GP-50S Printer

Seikosha was a branch (absorbed in 1996) of the Japanese company Seiko. In the 1980s it also produced low-cost printers.

For the Sinclair ZX81 and ZX Spectrum models, it brought to market the GP 50 S dot-matrix impact printer (fig. 10).

The writer was delighted to receive one as a present from his parents as a school prize, and used it to its full potential.

The power supply was separate

and the cable to the computer a bit short. However, the connector left room for adding other peripherals. It was suggested to connect the printer and switch it on before switching on the computer.

The printer cartridge contained a 40-cm ribbon refreshed by an ink pad, and a roll of ordinary calculator paper at least 3.5 inches wide could be used.

The printer could print 32-column listings and COPY graphs (although, for example, circles appeared oval). Being also slow and noisy, it was certainly not a printer for professionals, but it was ideal for a student with a ZX Spectrum.

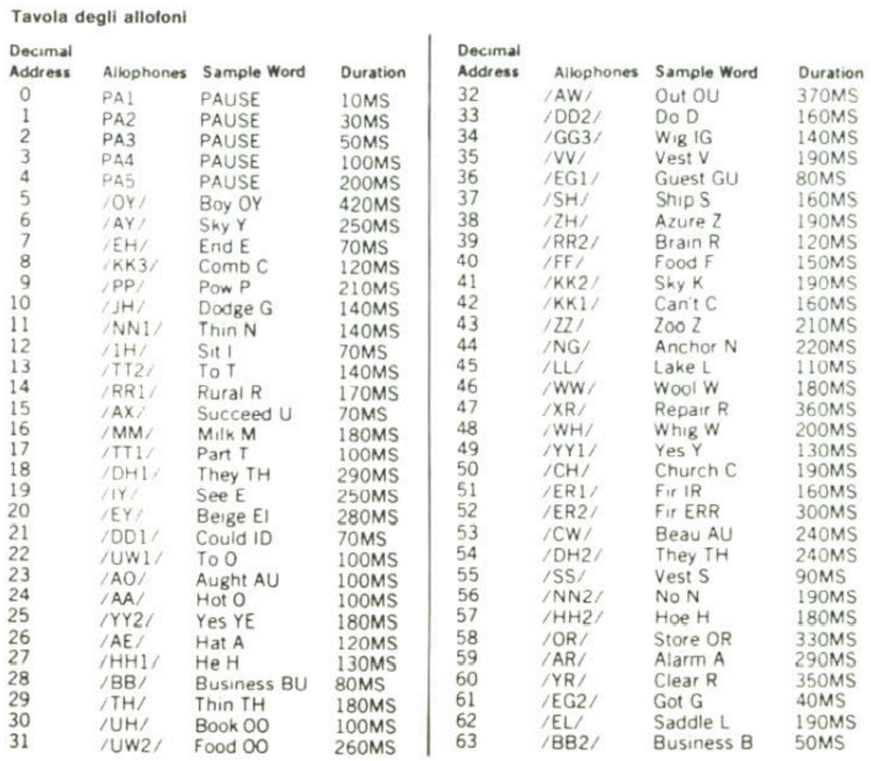


Tavola degli allofoni

Decimal Address	Allophones	Sample Word	Duration	Decimal Address	Allophones	Sample Word	Duration
0	PA1	PAUSE	10MS	32	/AW/	Out OU	370MS
1	PA2	PAUSE	30MS	33	/DD2/	Do D	160MS
2	PA3	PAUSE	50MS	34	/GG3/	Wig IG	140MS
3	PA4	PAUSE	100MS	35	/VV/	Vest V	190MS
4	PA5	PAUSE	200MS	36	/EG1/	Guest GU	80MS
5	/OY/	Boy OY	420MS	37	/SH/	Ship S	160MS
6	/AY/	Sky Y	250MS	38	/ZH/	Azure Z	190MS
7	/EH/	End E	70MS	39	/RR2/	Brain R	120MS
8	/KK3/	Comb C	120MS	40	/FF/	Food F	150MS
9	/PP/	Pow P	210MS	41	/KK2/	Sky K	190MS
10	/JH/	Dodge G	140MS	42	/KK1/	Can't C	160MS
11	/NN1/	Thin N	140MS	43	/ZZ/	Zoo Z	210MS
12	/IH/	Sit I	70MS	44	/NG/	Anchor N	220MS
13	/TT2/	To T	140MS	45	/LL/	Lake L	110MS
14	/RR1/	Rural R	170MS	46	/WW/	Wool W	180MS
15	/AX/	Succeed U	70MS	47	/XR/	Repair R	360MS
16	/MM/	Milk M	180MS	48	/WH/	Whig W	200MS
17	/TT1/	Part T	100MS	49	/YY1/	Yes Y	130MS
18	/DH1/	They TH	290MS	50	/CH/	Church C	190MS
19	/IY/	See E	250MS	51	/ER1/	Fir IR	160MS
20	/EY/	Beige EI	280MS	52	/ER2/	Fir ERR	300MS
21	/DD1/	Could ID	70MS	53	/CW/	Beau AU	240MS
22	/UW1/	To O	100MS	54	/DH2/	They TH	240MS
23	/AO/	Aught AU	100MS	55	/SS/	Vest S	90MS
24	/AA/	Hot O	100MS	56	/NN2/	No N	190MS
25	/YY2/	Yes YE	180MS	57	/HH2/	Hoe H	180MS
26	/AE/	Hat A	120MS	58	/OR/	Store OR	330MS
27	/HH1/	He H	130MS	59	/AR/	Alarm A	290MS
28	/BB/	Business BU	80MS	60	/YR/	Clear R	350MS
29	/TH/	Thin TH	180MS	61	/EG2/	Got G	40MS
30	/UH/	Book OO	100MS	62	/EL/	Saddle L	190MS
31	/UW2/	Food OO	260MS	63	/BB2/	Business B	50MS

Fig.9

SEIKOSHA GP 50 S

Printing mode: dot-matrix impact (8x8)
Head: single hammer
(30 mln characters guaranteed)
Format: 32 columns
Line spacing: 1/9 inch
Speed: 35 characters per second (declared)
Horizontal space between dots: 1/84 inches
Vertical dot spacing: 1/72 inches
Paper feed: 6.6 lines per sec.
Drag: friction drive
Power supply: 220V AC-18V DC transformer
Power consumption: 11W in stand-by,
17W during printing
Dimensions: 85mm x 250mm x 215mm
Weight: 1.5 Kg

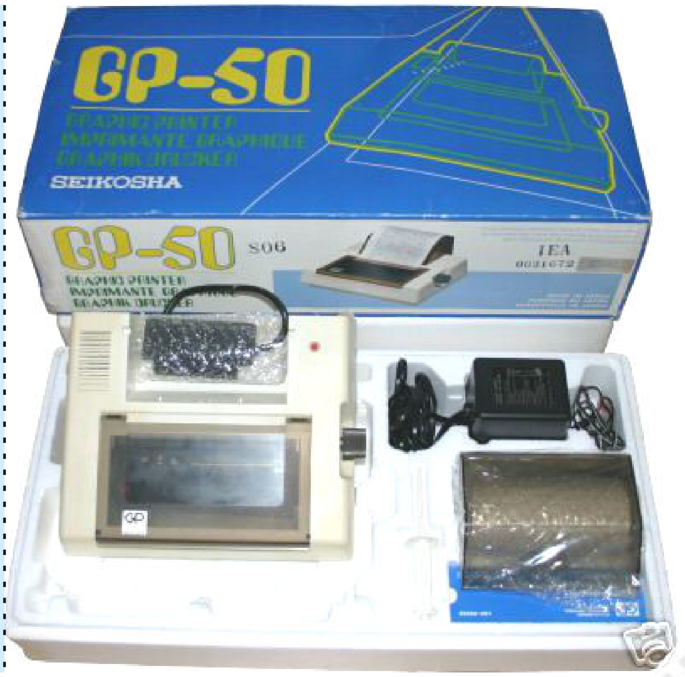


Fig.10

Bibliography

[Dis21] AA.VV., "DISCiPLE", retrieved (2021-12-10) from https://en.wikipedia.org/wiki/DISCiPLE

[OD21] Spectrum Computing retrieved (2021-12-10) from https://spectrumcomputing.co.uk/index.php?cat=96&id=1000297

[Err85] G. Errichelli, "Floppy Disk Sandy", Sperimentare con l'elettronica e il Computer, Jul-Aug 1985, pp. 40-43, http://www.introni.it/pdf/Sperimentare%201985_07_08.pdf

[Ber84] M. Bergami, "CHEETAH Sweet Talker", MC Microcomputer n.29, Apr. 1985, pp. 68-70, http://www.dizionariovideogiochi.it/doku.php?id=mc_-_029

[SC84] AA.VV., "Seikosha GP50S", Sinclair Computer n.7, Oct 1984.

[SCo21] AA.VV., "Disciple Interface", retrieved (2021-12-10) from https://spectrumcomputing.co.uk/entry/1000117/Hardware/DISCiPLE_Interface

[Spe85] Sperimentare Dec. 1985, pp 8-9, retrieved (2021-12-10) from http://www.introni.it/pdf/Sperimentare%201985_12.pdf

[Wa87] J. Wase, "Disciple", ZX Computing Monthly, Mar. 1987, pp.30-31.

[WS21] World of Spectrum retrieved (2021-12-10) from https://worldofspectrum.org/hardware/

ColecoVision

by Leonardo Miliani

Following the thread of the consoles that have marked the industry more than others, in this article we are going to analyze an object belonging to the second generation of home gaming devices, born, like the Fairchild Channel F analyzed in the previous issue, on the American East Coast. This console holds a couple of records: it is the first of its kind that has offered games with graphics equal to those of the arcades of the time, and it is the first that has offered hardware expansions that also radically change its basic features. We are talking about the ColecoVision (fig. 1).

A bit of history

To understand the genesis of the devices analyzed in previous articles, we almost always started several years before their arrival on the market but this time we exaggerate, and not a little: this time we go back to 1932! We are in West Hartford, a town of a few tens of thousands of people in central Connecticut. Maurice Greenberg founds the Connecticut Leather Company, a company that sells hides and tools to cobblers who repair shoes. In 1938, he begins producing and selling leather boots as well. With the entry of the United States of America into World War II, business takes off due to the increased demand for these items. In 1945 the sales offer is rich: machinery for the processing of shoes, hats and accessories for their cleaning, and stations for shoe shiners. Within a few years, thanks to the ideas of his son Leonard, Greenberg's company also began to offer kits for making its own leather goods, moving from wholesale to retail.

But what does this have to do with gaming consoles? It is precisely the embrace of sales to the final consumer that will radically change the company's market sector within a few years. In 1954, in fact, during the New York Toy Fair, an event dedicated to the world of toys, the kit for making leather moccasins wins the prize for the best educational toy for children! Thanks to this award, Leonard Greenberg shifts the company's interests towards the toy sector, eventually selling the leather goods branch of the business in 1961 and changing its name to Coleco Industries Inc. In 1963, Coleco entered the world of above ground swimming pools, becoming the largest manufacturer in the world through the acquisition of a major company in the industry. In 1966 Leonardo's brother, Arnold, joined the company, suggesting further acquisitions and guiding the company towards success in the toy sector. At the beginning of the 70's Coleco enters the snowmobile market, but business is not going very well and profits drop. Arnold Greenberg, who became CEO, in 1976 decides to enter the world of game consoles attracted by the success of Atari's Pong and all its clones, which have literally invaded the homes of Americans.

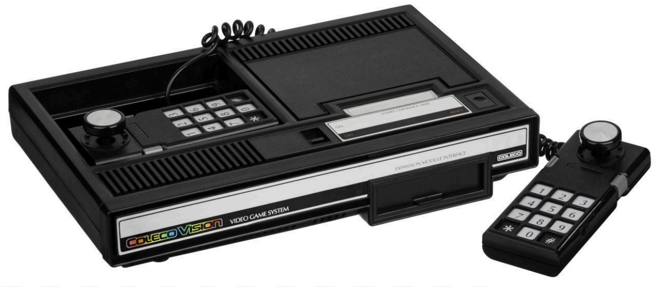


Fig. 1: ColecoVision (photo: Evan-Amos – source: Wikimedia Commons)

The ancestors of ColecoVision

In 1976 Coleco released the Telstar (fig. 2), a simple console to be connected to the TV set with 2 integrated paddles based on the General Instruments "Pong-on-a-chip", a single integrated offering 3 games in Pong/Tennis style, which became an immediate success: more than 1 million units were sold. However, interest in these devices quickly wanes, overshadowed by the new programmable consoles coming out at the end of the 1970s, first and foremost the Atari VCS/2600. This led Coleco almost to bankruptcy but the company managed to recover by producing small portable video games such as the Electronic Quarterback, a game dedicated to football. Thanks to the quality of these video games, Coleco steals the first place to Mattel, which until then was the undisputed leader of that market niche. Strengthened by this new success, in 1979 Coleco began the development of a real home console. Eric Bromley, an engineer who had previously worked for other arcade manufacturers including Midway, was hired to develop a cartridge-based system as a means of distributing games. The quality of the system is very good but the hardware requirements are considerable and Greenberg stops the development because the large

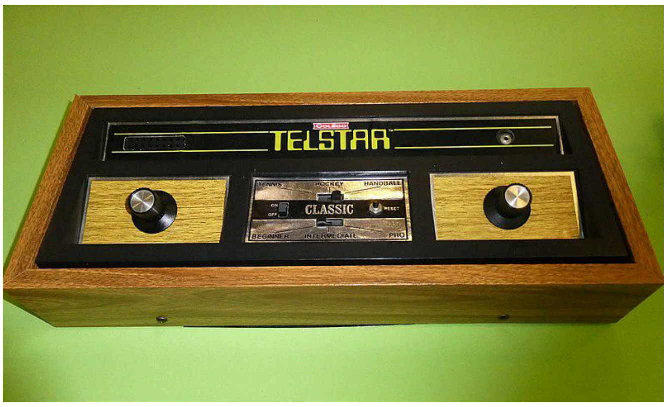


Fig. 2 - Coleco Telstar (photo: Jzh2074 – source: Wikimedia Commons)

amount of RAM required makes the production costs too high (do not forget that at the end of the '70s of last century RAM was very expensive so much so that the first home computers had small allocations of memory, in the order of 1, 2, or 4 KB on average). Things changed in 1981: RAM prices started to fall and Bromley came back to Greenberg showing that it was now possible to make his console and put it on the market at a competitive price. Greenberg authorizes the resumption of the development of the device, but he also understands that the console will need a strong driving force to enter the fray, at the time dominated by the Atari 2600 and Mattel's Intellivision. Coleco, which in the meantime had continued to invest in the video game market, at the beginning of 1982 put on the market the Mini-Arcade, small reproductions of the most popular arcade games of that period but based on solid state electronic components (like the first generation consoles) that reproduce (in a very limited way) the game mode of the replicated arcade, be it Pac-Man, Frogger or Galaxian. These objects have a huge success, with sales in the millions of units. One of the factors in this success is the fact that they are able to license the names of the most popular arcade titles. Greenberg and Bromley decided to bundle the console with an exclusive hit game, not only to show off its graphics capabilities but also to force those who wanted to play the game at home to buy the console.

Negotiation with Nintendo

At that time the biggest arcade games came from Japan and Greenberg, in early 1982, sent Bromley to Kyoto to speak directly with Nintendo president Hiroshi Yamauchi to negotiate exclusivity for a successful title. Negotiations go on with ups and downs: Yamauchi proposes games of little value, that surely wouldn't attract much attention. But luck is about to put a patch in it: during a lunch break, going away for a simple physiological need leads Bromley to the discovery of gold! While he is looking for the bathroom, he passes in front of a room where there is a cabinet with a big monkey on the sides and a game that immediately attracts him: "Donkey Kong" (fig. 3). That's the game Coleco needed! Negotiations with Yamauchi immediately shifted to Donkey Kong and Bromley agreed to its exclusivity: the deal called for the immediate payment of $200,000 in license fees and a $2 royalty for each cartridge sold. Bromley returns to America with an agreement signed on paper napkins... because at that time in Japan contracts were mostly verbal agreements. But things are about to take a turn for the worse. During the Consumer Electronic Show in June, Bromley discovers that Donkey Kong is being shown at the Atari booth! Bromley learns from Yamauchi's daughter, who is acting as interpreter during the negotiations with her father, that the president of Nintendo, wanting to enter the American video game market, has made an agreement with Atari, then one of the biggest names in the industry, to market his new game. Bromely, with the help of his daughter, manages to negotiate a new agreement (this time signed) by which Coleco is granted Donkey Kong exclusively for consoles, while Atari retains the rights for computers.

Marketing and expansion modules

The new console is put on the market during the month of August 1982 as ColecoVision at the launch price of $ 174.99: each package contains the game Donkey Kong inside. Other games then follow the launch of the machine. Coleco advertises in magazines and newspapers the technical qualities of the ColecoVision, far superior to those of its direct rivals Atari 2600 and Mattel Intellivision: for the first time the conversions are very faithful to the

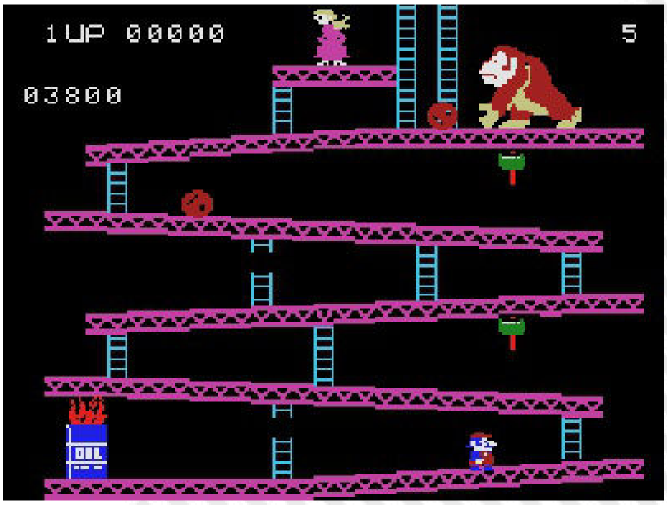


Fig. 3 - Donkey Kong

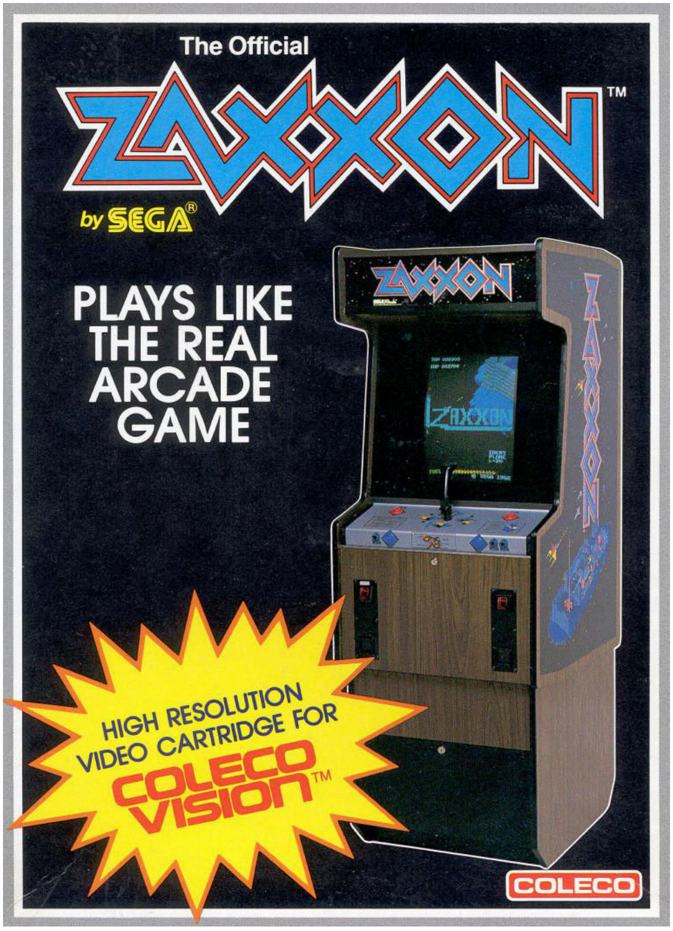

The Official
ZAXXON™
by SEGA®
PLAYS LIKE THE REAL ARCADE GAME
HIGH RESOLUTION VIDEO CARTRIDGE FOR
COLECO VISION™
COLECO

Fig. 4 - Cartridge cover of the game "Zaxxon". Note the emphasis placed on the fact that the game is faithful to the original (photo: Servo - source: Mobygames.net)

arcades of pubs and amusement arcades (fig. 4). Donkey Kong is practically the same as the original, Lady Bug, Turbo and Smurf (the "Smurfs") surprise, Zaxxon, with its isometric perspective, impresses everyone.

All this is possible because the ColecoVision is the first console to be based on a CPU normally used on computers: in fact it uses the established Zilog Z80 that drives the vast majority of 8-bit machines with CP/M operating system, which offers a computing power far superior to the "electronic brains" that move the other consoles. Alongside it is the TMS9928A video chip, a variant of the better known TMS9918A that is used in the Texas Instruments TI-99/4A and that later became the heart of MSX systems and other computers and consoles, which offers an image of 256x192 pixels in 16 colors with the beauty of 32 sprites with resolutions from 8x8 to 32x32 bits. The sound is generated by Texas Instruments SN76489A, widely used in computers, consoles and arcades: it offers 3 audio channels and 1 white noise channel. On the memory side, the system is richly equipped: 8 KB of ROM containing the BIOS of the machine, 1 KB of RAM for program data and 16 KB of VRAM exclusively dedicated to the video processor to store image and sprite data. The design of the console is very neat: it is a single black plastic assembly low and sleek, with a top slot to store the gamepads when not in use. The latter are equipped with a joystick and a numeric keypad used for additional commands of the various games, on which are applied captions provided by the manufacturers of the cartridges to more easily remember the meaning of the commands. One note that stands out in all of this is the inexplicable 12-second delay, from the time the console is turned on, before the game actually starts, during which the console only shows the "ColecoVision" logo on the screen: this is not a defect, it is a precise (and unwise) commercial choice by Coleco (when it is introduced on the European market, this delay will be reduced to 3 seconds). However, some game manufacturers find a way to bypass this BIOS delay and start their games immediately. By the end of the year Coleco sold 500,000 consoles, which became 1,000,000 in early 1983. In that year other titles arrive such as Buck Rogers, Centipede, Defender, Donkey Kong Jr., Frogger, Galaxian, Pitfall!, Popeye, Time Pilot and others, followed in 1984 by other successful titles such as BurgerTime, Choplifter, Gyruss, Q*berts, River Raid, Spy Hunter. Sales increase with the beginning of the distribution outside the USA: in July 1983 the console makes its debut in Europe, distributed by CBS Electronics as "CBS ColecoVision".

What makes the ColecoVision attractive are its "Expansion Modules", accessories that are generally connected to the expansion port of the console that also substantially modify its characteristics. The "Expansion Module #1" transforms the ColecoVision into an Atari 2600 thanks to the fact that inside there is a perfect clone of this console. This module allows the ColecoVision to access the vast library of Atari 2600 games. Atari also sued Coleco in court for this accessory but lost the case because at the time the American laws on patents on video games were a bit gaps and since the module was made using components commonly available on the market, the verdict was that according to the law no proprietary component had been used without a license (thanks to this judgment, Coleco also markets the adapter as a stand-alone console under the name of "Coleco Gemini"). The "Expansion Module #2" (fig. 5) is essentially a steering wheel with pedal box initially offered with the "Turbo" cartridge. The gas pedal is a simple on/off switch, thus offering no analog

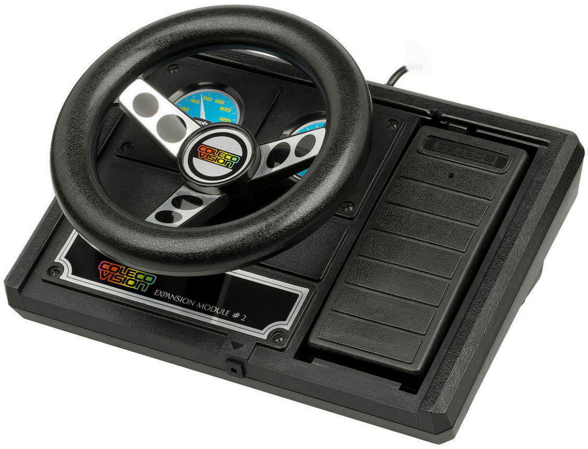


Fig. 5 - Expansion Module #2
(photo: Evan Amos - source: Wikimedia Commons)

gas control. It has also been used by other driving games but has not had much commercial success. "Expansion Module #3" is the most interesting expansion. Since in the first half of the '80s home computers are becoming more and more popular, Coleco decides to ride the wave and, strong in the fact that its console integrates an 8-bit CPU normally used on computers, creates this expansion that transforms the ColecoVision in the "Adam Computer" (fig. 7). It consists of a machine body to which the console is attached at the back and which contains, in addition to the motherboard with 64 KB of RAM, a magnetic tape reader, a printer, a keyboard and the cartridge of the game "Buck Rogers: Planet of Zoom" (it cannot use Donkey Kong because the rights for this game on computers, as written above, are held by Atari). A disc drive can be purchased separately. Despite its potential, the various flaws that plague it and the high cost limit its success.

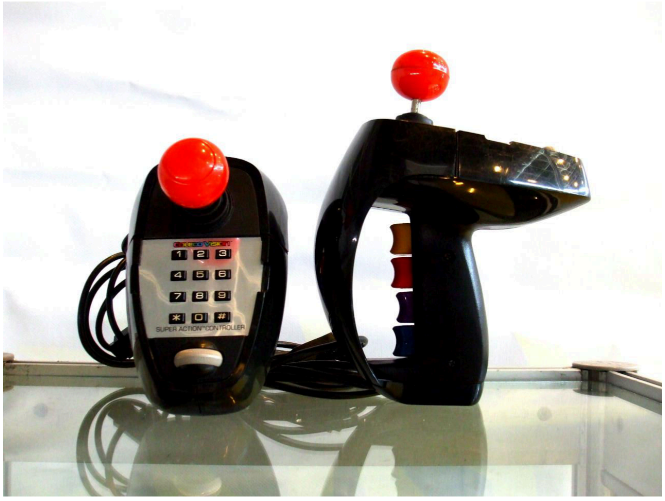


Fig. 6 - Super Action Controller
(photo: Miguel Duran - source: Museo 8 bits)

In addition to these modules, advanced controllers are also offered. The "Roller Controller" is a trackball offered with the game "Slither" which, absurdly, uses a connector that is not compatible with the Adam computer so Coleco is forced to ship the adapter to those who request it. The "Super Action Controller" (fig. 6) is instead a set of 2 special joysticks initially offered with the "Super Action Baseball" cartridge. Each device has a joystick with a ball top on the top, where there is also a keyboard with 12 buttons and a speeder roller, a roller potentiometer to control the speed of the ball; in the handle there are 4 additional buttons. This special controller has also been used by other games.

It should be noted that originally Coleco had begun the design of a module called "Super Game Module" (SGM), whose development had participated in a second time also Ralph Baer, the creator of the first console in history, the Magnavox Odyssey. The module was supposed to allow games to be loaded from a tape or disc player. The idea was shelved when it was decided to develop the Adam module.

The 1983 crisis and the exit from the scene

The console, as mentioned, sells really well. The quality of its games, strengthened by the hardware capabilities of the machine, is superior to that of its rivals and its conversions are truly comparable in appearance and playability to the arcades from which they derive. What ruins a little bit the image of ColecoVision are the vaporware from which Coleco suffers: too many times games or products are presented that then undergo considerable delays or are even cancelled and never put on the market. One of these vaporware is the already mentioned SGM: announced for the month of August 1983, it was first delayed, then revised, and later cancelled, replaced by the Adam module. This also suffers the same fate: presented in June 1983, was to arrive soon and sell, according to plans, 500,000 pieces by Christmas. The output is however postponed month by month, and the Adam arrives on the market only in mid-October, with a price increased from the initial 525 dollars to 725 final. If we add to this amount the cost of purchasing the console, we arrive at a figure higher than that necessary to purchase the ColecoVision and a Commodore 64, the computer that in that period is imposing itself in a massive way for its graphic and sound qualities. The Adam is then plagued by serious manufacturing defects: many units fail after a short use, others are even defective already at the factory, with the

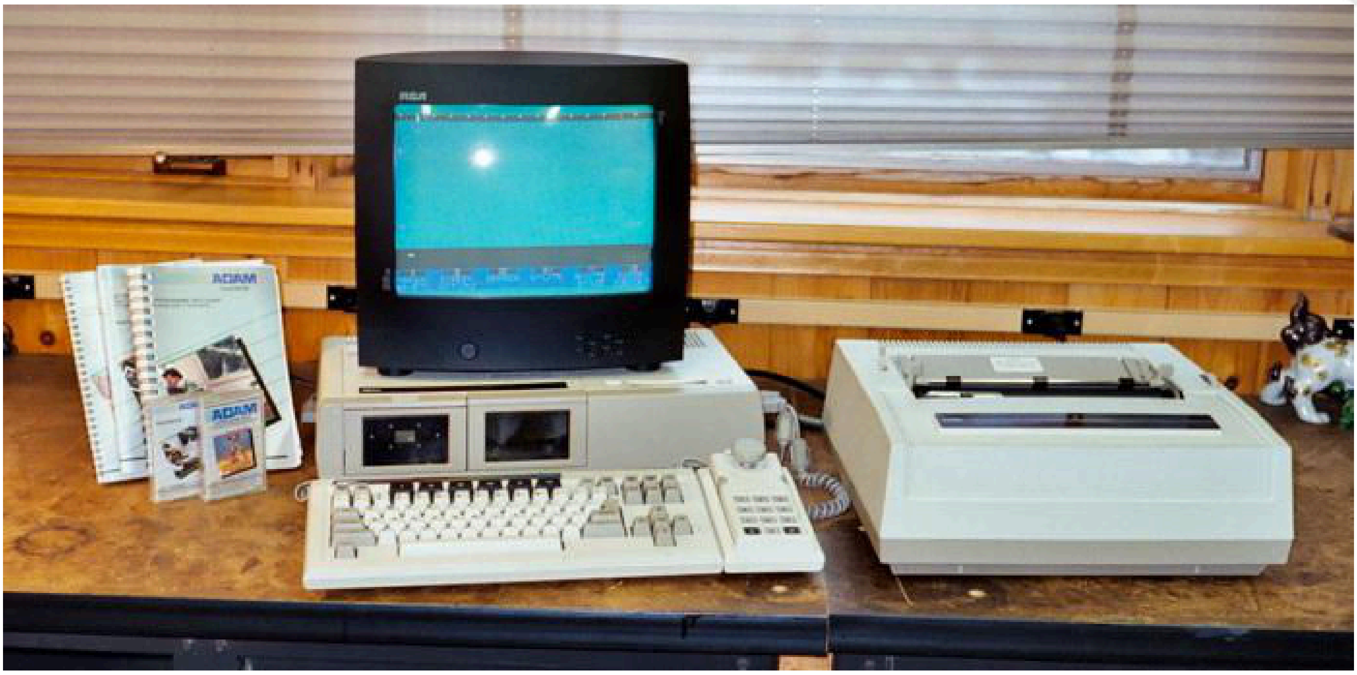


Fig. 7 - Coleco Adam (photo: Doontass - source: Wikimedia Commons)

result that the returns for repairs are high. Because of this by 1983 only 100,000 units are sold, with an estimated loss of $35 million.

In 1983 another event occurred that undermined Coleco's finances: the famous "video game crisis" took place, which hit the industry hard, leading smaller companies to bankruptcy and larger ones to a massive downsizing. Atari itself, which until then had dominated the video game sector, suffered losses of half a billion dollars that year, risking closure. Coleco also suffers the crisis: although the ColecoVision continues to sell reasonably well (it reaches 2 million units in 1984), the losses caused by delays and defects of the Adam and the poor sales of cartridges due to the crisis lead the parent company to re-enter the toy market to try to make money. At the end of 1983 it presents a series of cloth puppets called "Cabbage Patch Kids", which sell well and manage to keep the company afloat. 1984 Ends with another 80 million dollars in losses, mainly due to problems with the Adam: at the beginning of 1985 the company finally decides to take it off the market while by the end of the year it also stops selling the ColecoVision. The total sales numbers of the console are not known exactly, but it is estimated that a total of 6 million ColecoVision were marketed: figures are not striking but certainly interesting, given the period in which the console was put on sale, having had to ride the wave of the crisis in the industry that hit America.

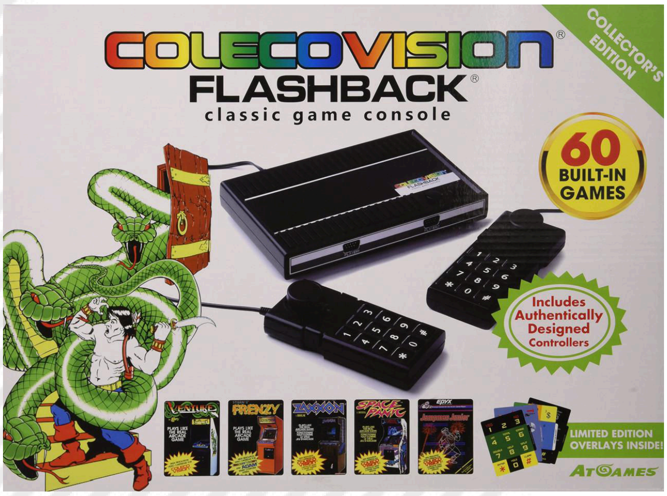


Fig. 8 - ColecoVision FlashBack (source: AtGames)

In more recent times, the console has experienced a second youth thanks to several independent developers who have released new games and, thanks to the similarities with the MSX and Sega SG-1000 systems, made the porting of numerous games from these two platforms. Not only that, in 2014 the "ColecoVision FlashBack" (fig. 8) was put on sale, a modern version of the console with 60 pre-installed titles to let new generations savor the taste of games from the 80s: a way to honor a system that could have had so much to offer but did not materially have the time.

RetroFighter engine for Commodore 64

by Gianluca Alberico

My name is Gianluca Alberico, I'm 42 years old, from Rome, diploma of industrial engineer. I'm a "congenital" programmer: I've been programming since I was 10 years old, I started with a C64, then Amiga and finally a PC clone. I have always had a passion for programming video games and computer graphics in general. After graduation I created my own website, with some games developed by me for MS-DOS. With the passing of the years and due to some life choices, I then became a .NET/Xamarin programmer, but in my spare time I'm always doing something related to game programming.

The reasons that pushed me to create the engine

Towards the end of 2021 I had been wanting to play some C64 fighting games, even those recently released, but I realized that there weren't any yet that presented a playability equal or similar to those that generally came out for the consoles of the time (SNES, Genesis, etc.). I had also noticed the YouTube demo of SF2CE by Pacoblog64 [1], but I was not satisfied with it. So at the beginning of October 2021, after some research, I set myself a technical challenge: is it possible to create a fighting game engine on the C64 that does something more than what we have seen so far? And so I started to develop an engine, initially for PC, in C language, that would reproduce first the playability of Street Fighter 2 but with sprites suitable for conversion on C64. Then I took to converting both graphics and code for C64 and, solving one problem at a time, I was able to produce what is already in my channel [2].

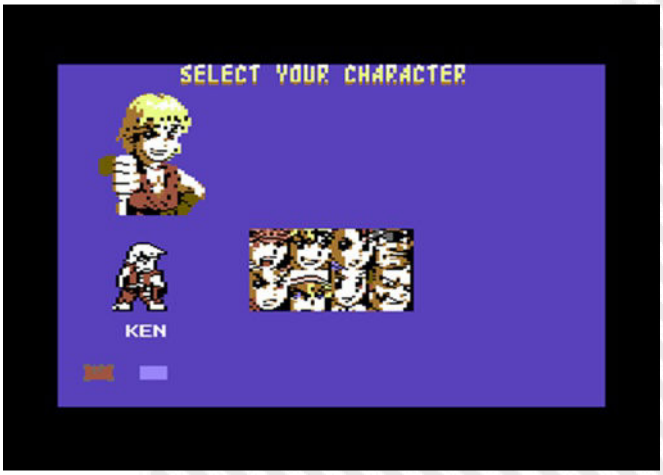
What does the engine consist of?

The engine is able to implement a combat session within a fighting game between two players. The heart of the engine, that is what represents the real game session, is developed in C language to allow a future portability of the engine on other platforms (e.g. Amstrad CPC). The structure of the engine reminds a bit the MUGEN [3] one, but obviously its potentialities are very reduced in comparison to this last one because it must be executed on 8 bit systems (for now) with limited hardware resources.

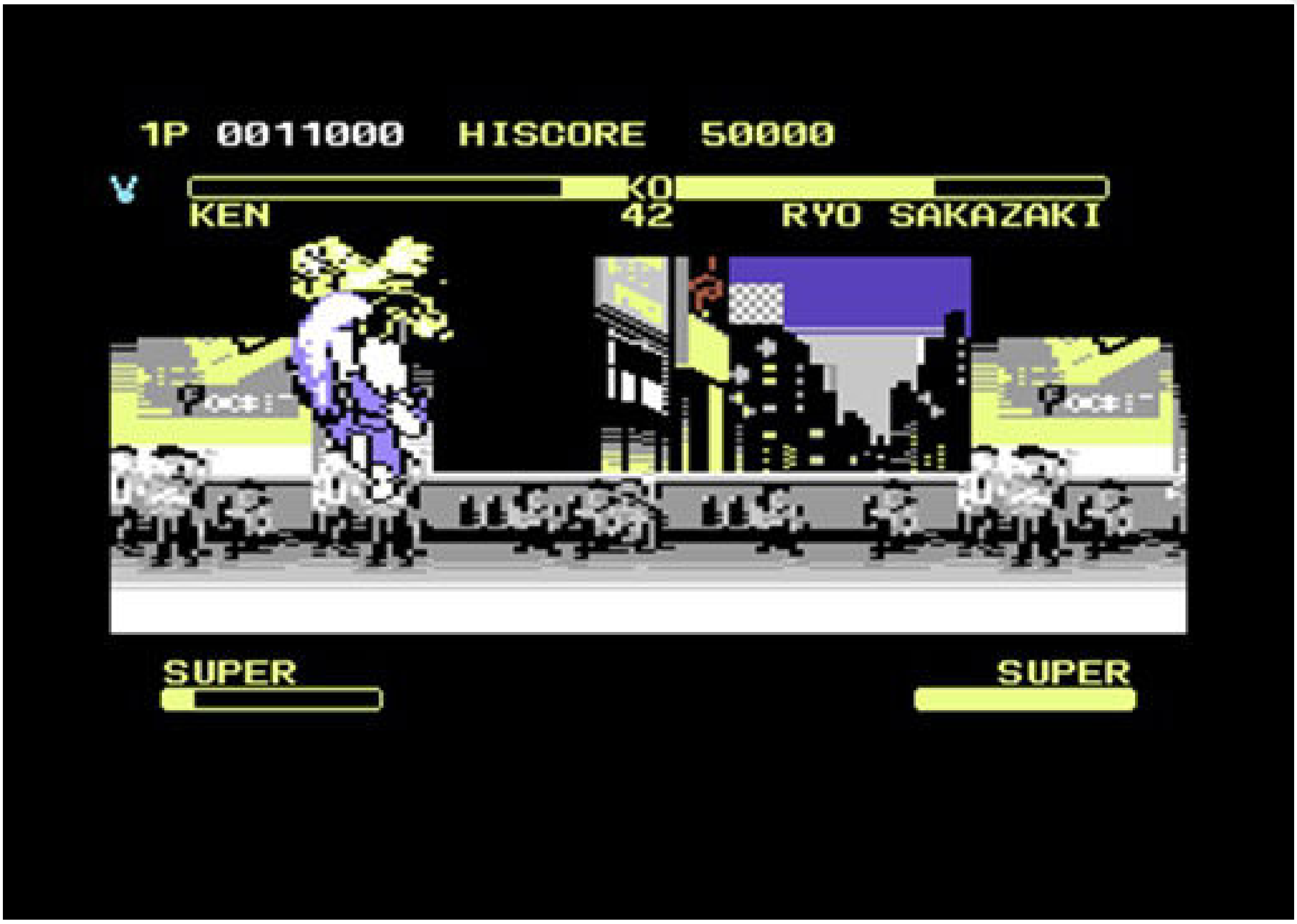
What it can do

<u>The engine has the following features:</u>

- Each fighter can draw up to 63 frames of different size. There are no drawing constraints within the frames (they can be drawn as one would draw the player frames of a similar PC game), the important thing is to respect the C64 limitation of having 3 colors plus 1 for transparency. The engine implements a sprite multiplexing technique to allow the extension of sprites vertically; however, to limit memory consumption, the engine currently allows a maximum frame size of 48x63 pixels.
- Each fighter is defined by a collection of sequences. Each sequence is a script that can execute several commands (move, throw projectiles, some effects, etc.). There are currently almost 40 sequences that can be defined for each fighter, but in a while I may be able to define more (about 60) and especially longer ones.
- Each fighter can currently perform 6 basic ground moves, 2 flying moves, 3 special moves, and one super move. In a little while, I may be able to increase the number of basic, special, and super moves to perform.
- Each fighter can only perform 1 type of grapple. As before, in a while, I may be able to increase the number of holds I can perform.
- Each fighter can perform the blows most commonly found in all fighting games: weak and strong blows, stunning blows, immobilizing blows, and flaming blows.
- Backgrounds have a fixed size of 320x112 pixels, and must be character-based (they cannot be drawn freely



Character selection with its color



Flaming blows and special moves

pixel by pixel but must be composed of a series of blocks from a custom palette). Backgrounds can be animated: simple animations of course, which consist of simply changing the content of a part of the screen with other content periodically.

- Currently they are not implemented but in the next demo there will be the possibility to create very simple bonus rounds, in which the player will have to destroy one or more animated or non-animated objects.
- Two "movies" can be defined for each fighter that will be shown in the story mode: introductory and final. The introductory one will be shown after the character selection, the final one will be shown after the last opponent is beaten. Each movie is also a script and can draw from different bitmaps (so it can contain whatever you want, without being bound to blocks but respecting the C64 color restrictions). The same scripting language can be used to create the initial animation of the game when it is started.
- For each fighter it is possible to define the list of his opponents.
- For each fighter it is obviously possible to define the list of characteristic phrases of victory and provocation.
- It is possible to define the colors that the player can choose for each fighter.

What it can't do

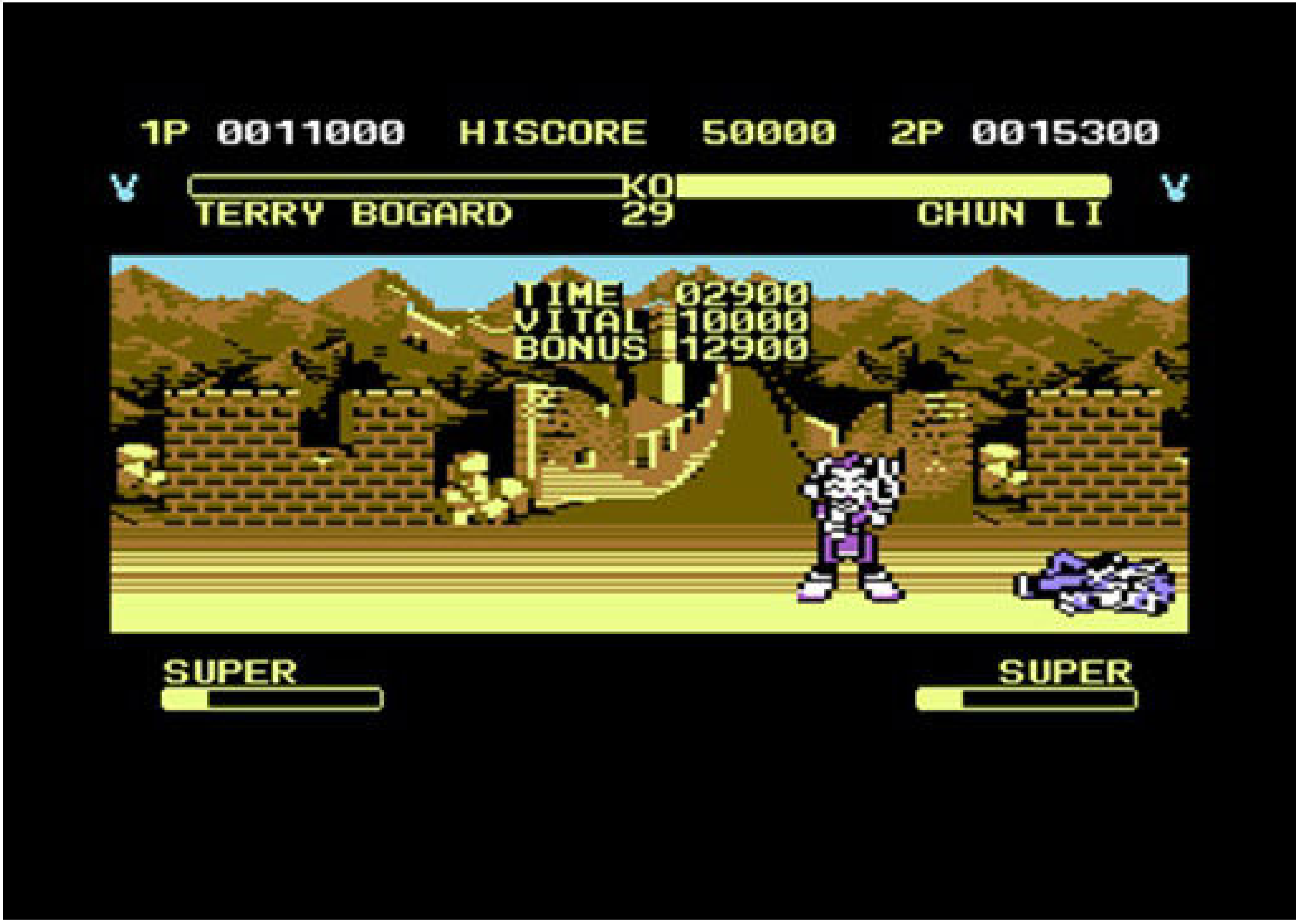
Currently:

- The engine does not provide an internal engine to play background music but can only play simple sound effects.
- The engine is unable to define special flying moves.
- The engine is not capable of allowing a player to shoot more than one bullet at a time.

Definitely, in the future:

- The engine will not be able to use bitmap-based backgrounds as, if it did, the background animations would be slower and the frame rate too low.
- The engine only provides for a narrow field of play that is a maximum of 320 pixels wide.

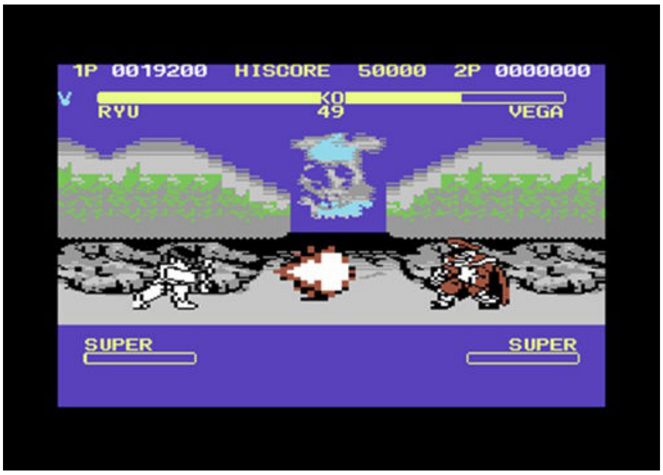
By saying "at the moment" I don't necessarily mean that in the future a certain feature will be implemented, but only that I will consider it for further evaluation (except of course the music background that will be included in the game to be produced with this engine).



Several stages, animated and not

A game to complete: SNK vs CAPCOM

My intention is to create a game called simply "SNK vs CAPCOM", inspired by the game of the same name that was released for Neo Geo Pocket Color. The game will not be a true copy of the NGPC game, but simply a customized version adapted for the C64. It will simply be a game along the lines of Street Fighter 2 but with the characters of SNK vs CAPCOM: a tournament for many fighters to participate, each with their own motivation. In the game there will be 13 characters (12+1 secret). The game will be released in cartridge format (.CRT file).



Super special moves for all characters

How do you integrate?

At the moment the engine is not usable by other programmers because it still has to be completed and the single constituent parts have to be separated better. However, in the future I will seriously consider the possibility of making it become an SDK that can be integrated on different platforms.

References

[1] SF2CE PacoBlog64: https://www.pacoblog64.com/2020/05/desarrollo-de-street-fighter-2-champion.html

[2] RetroGL - https://www.youtube.com/channel/UCshZEsIkWs_xUjXdifb3T4A

[3] MUGEN Engine - https://mugen.weboy.org/category/database/full-games/

Atari ST Basic - AES and VDI

by Francesco Fiorentini

As a boy, before moving to PCs, I owned the Commodore 64 and the Amiga 500. Like many of our readers I started programming in Basic with the Commodore 64 then in AmigaBASIC on the Amiga, and then I moved to other languages on the 16-bit Commodore and IBM PCs.

The Basic language has always fascinated me for its simplicity of use and especially for the fact that it was, almost always, the default language on 8 and 16 bit machines. It was therefore natural, at the time, that I wondered what environments were present on machines I did not own and was curious to try them out. Unfortunately, not being the son of Kashoggi, the funds available at the time did not allow me to afford more than one computer per time, so I always had to keep my curiosity under control. Until now...

Now, with emulators, I can indulge myself and try to use computers I didn't own and, above all, write two lines of code in a Basic that I couldn't use at the time. Among all the computers I have owned, I have always had a weakness for the Amstrad CPC and the Atari ST. I satisfied my curiosity with the Amstrad by using Locomotive Basic and discovering an incredibly advanced language compared to the poor Basic V2. My experiments with that machine and that language aren't finished, but my curiosity is now pushing me to try the Atari ST and its ST Basic. From what I've read, the functionalities of the ST Basic are unfortunately comparable to those of the miserable Amiga Basic, so no exciting discovery like with the Locomotive Basic. You may be wondering why then waste time trying this language. The answer is simple, it's part of the history of the machine and serves to understand the reason why other Basic (GFA Basic, STOS...) were created for this system (...and for the Amiga). And then, as I always say, it's fun to discover new things.

Atari ST Basic

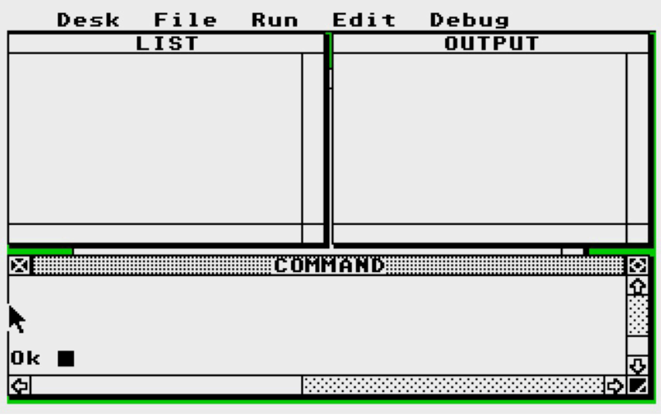
The first release of the ST Basic was in 1985, the same age as TOS. Produced by MetaComco at Atari's request to exploit the GEM environment on which the Atari ST is based (we will see later what it is about), it turned out to be unfortunately full of bugs and therefore inadequate for the ambitions of the host machine. Since it was supplied with the Atari ST, it obtained however a discreet audience since, at least in the beginning, magazines and software producers used it to distribute elementary applications.

ST Basic - development environment

The development environment of the ST Basic is very similar in concept to that of the Amiga Basic, so anyone with a minimum of familiarity with that environment will soon find themselves at home.

The interface consists of 4 windows:
- **EDITOR** - where to insert the actual listing
- **LIST** - displays the entire list of the programme
- **COMMAND** - where to insert the immediate commands (and also the listing)
- **OUTPUT** - where the result of the programme is shown

Fig. 1 - Atari ST (picture from Wikimedia Commons)



```
Ok ■
```

Fig. 2 - ST Basic at start

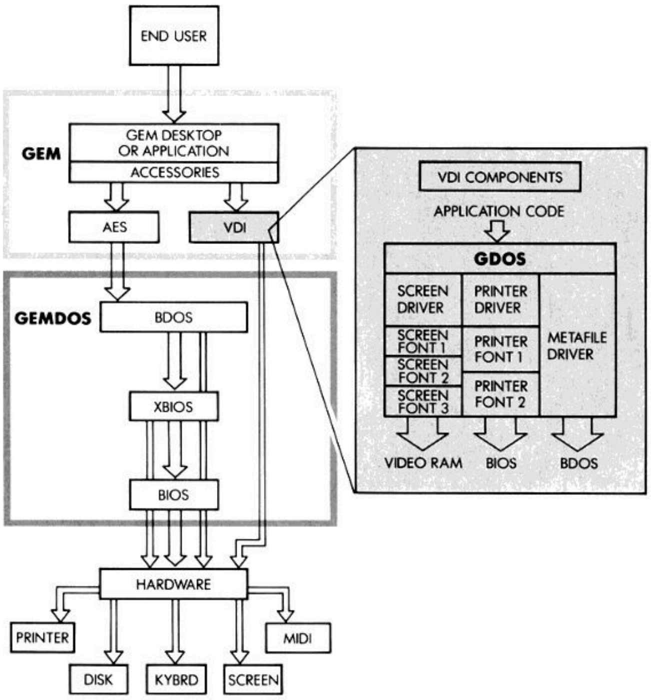


Fig. 3 - TOS Logical Architecture

When ST Basic is started, three of the above-mentioned windows are displayed at the same time (figure 2). Given the small size of the screen, this is not exactly the best, but with a little practice we will learn how to navigate.
If I have to make an immediate comparison with the Amiga Basic, I found the ST Basic environment slightly better performing and therefore less frustrating to use.

For obvious reasons of space and to make the article a bit more interesting, instead of talking about the usual commands, more or less in common with all Basic dialects, I found a very interesting article in the Atari ST User magazine that talks specifically about how to access, via ST Basic, some functions of the AES and VDI components of the operating system.

TOS, GEM, AES, VDI...

In computer science, acronyms have always been a must, and obviously Atari is not avoiding them as well. TOS, an acronym for The Operating System, is the operating system behind the ST series computers. Appeared in 1985 and developed by Atari Corporation, it was a windows and icons operating system not equipped with a real multitasking, but thanks to the possibility to load in memory Terminate and Stay Resident (TSR) programs, it simulated a cooperative multitasking.

The TOS, depicted in Figure 3, consisted of this set of components:

- **Desktop** - The main interface, loaded after start-up
- **GEM** - Graphical Environment Manager, consisting of:
- **AES** - Application Environment System, performs operations on the screen, such as managing windows, drawing and editing dialogue boxes, animating menus and many other things.
- **VDI** - Virtual Device Interface, which handles graphical I/O functions such as drawing lines and performing bit-blit operations
- **GEMDOS** - GEM Disk Operating System, also known as BDOS - Basic Disk Operating System, manages interactions with disks and files.
- **BIOS** - Basic Input/Output System, manages interactions with the hardware
- **XBIOS** - Xtended BIOS, handles hardware level calls proprietary to the ST, such as setting the paddle registers and configuring the audio chip.

VDI is again divided into three components: fonts, drivers and GDOS. Font defines font styles and sizes that can be written on any output device (video, printer...). Driver, as the word itself implies, is the part of the code that implements the virtual machine for a particular output device. The GDOS is the device-independent part of the VDI, which (among other things) sends VDI calls to the correct output device.

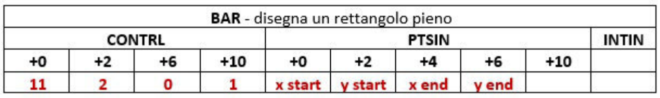
VDISYS()

The VDISYS() command is only briefly mentioned in the ST Basic manual. This is a pity, because it actually allows access to the VDI component and the use of some of its functions.

The functions can be invoked by passing parameters in three already defined variables: **CONTRL**, **PTSIN** and **INTIN** and then calling the function VDISYS(1). The value 1 passed to VDISYS is purely symbolic and irrelevant.
The functions that can be used are: **BAR**, **CHARACTER HEIGHT**, **CHARACTER SLANT** and **TEXT TYPE**. But let's see in detail how to do this.

- BAR function

The BAR function allows you to draw a solid rectangle and can be used to fill or clear entire areas of the screen very quickly.



BAR - disegna un rettangolo pieno

CONTRL				PTSIN					INTIN
+0	+2	+6	+10	+0	+2	+4	+6	+10	
11	2	0	1	x start	y start	x end	y end		

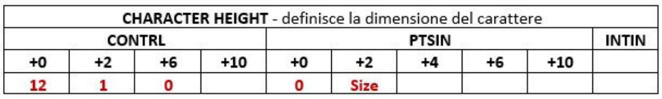
As can be seen from the table above, the parameters passed into the CONTRL variable are fixed and are used to invoke the BAR function, while PTSIN contains the start and end coordinates of the diagonal of the rectangle.

The sample programme will clarify all doubts once and for all.

```
10 fullw 2:clearw 2:colour 1,1,1,1
20 poke contrl,11
30 poke contrl+2.2
35 poke contrl+4.0
40 poke contrl+10.1
50 poke ptsin,50
60 poke ptsin+2.50
70 poke ptsin+4,150
80 poke ptsin+6,150
90 vdisys(1)
100 colour 1,1,2,24
110 poke ptsin+4,120
120 poke ptsin+6,120
130 vdisys(1)
140 colour 1,1,1,1
150 poke ptsin+4.80
160 poke ptsin+6.80
170 vdisys(1)
```

- CHARACTER HEIGHT function

As the name suggests, this function is used to manage the font size. Let's have a look at the parameters it uses.



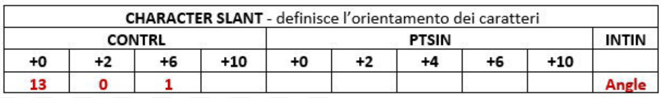
CHARACTER HEIGHT - definisce la dimensione del carattere									
CONTRL				PTSIN					INTIN
+0	+2	+6	+10	+0	+2	+4	+6	+10	
12	1	0		0	Size				

The font size is specified in PTSIN+2.

The default font size is 13.

- CHARACTER SLANT function

This function allows you to specify the orientation of the text. The orientation is specified in the INTIN variable and is expressed in tenths of a degree. Although it is possible to specify values from 0 to 3600, only values 0, 900, 1800 and 3600 will produce a result on the screen.

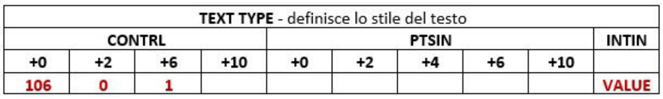


CHARACTER SLANT - definisce l'orientamento dei caratteri									
CONTRL				PTSIN					INTIN
+0	+2	+6	+10	+0	+2	+4	+6	+10	
13	0	1							Angle

- TEXT TYPE function

This function allows you to specify the style of text.

The text style is defined by the value specified in the INTIN variable in this way:



TEXT TYPE - definisce lo stile del testo									
CONTRL				PTSIN					INTIN
+0	+2	+6	+10	+0	+2	+4	+6	+10	
106	0	1							VALUE

0 - normal
1 - bold
2 - grey - grey
4 - italics - italic
8 - underlined
16 - outlined
32 - shadow - shaded

Let's take a look at a practical example that should clarify its use:

```
10 fullw 2:clearw 2:colour 1,1,1,1
20 poke contrl,106
30 poke contrl+2.0
40 poke contrl+6.1
50 poke intin,8
55 vdisys(1)
60 print "Francis
70 poke intin,4
75 vdisys(1)
80 print "Francis
85 poke intin,0
90 vdisys(1)
```

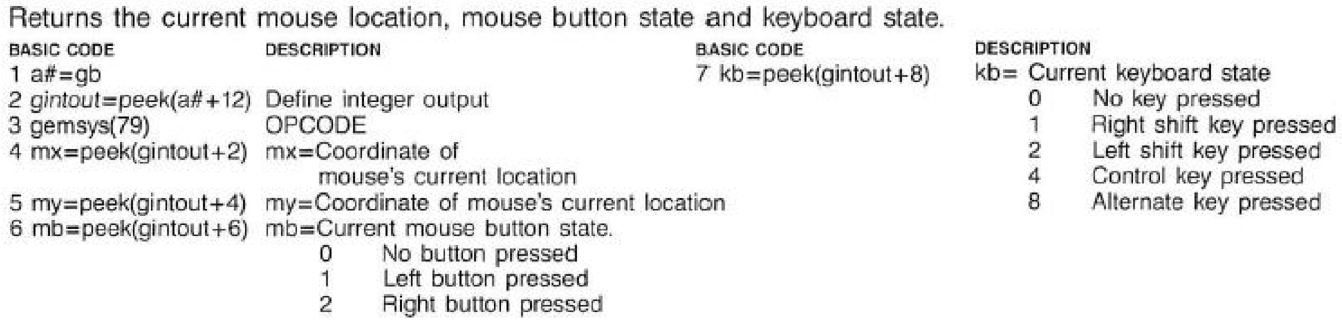
Of course, the effects can be combined simply by adding up the values. To obtain a bold italic, for example, simply pass the value 5 to INTIN.

Ah, remember to set the value back to 0 otherwise the text of the ST Basic will also continue to use the style you have set.

GEMSYS()

The command to invoke the AES functionality is GEMSYS. Just like the VDISYS command, GEMSYS requires parameters to specify the functionality to be activated. These parameters are passed to the AES in arrays called **CONTRL**, **INTIN**, **INTOUT**, **ADDRIN**, **ADDROUT** and **GLOBAL**. Well, what's the problem, we already know some of these data structures... No, it is not that simple. These data structures are not the same as the ones we used with VDISYS. They are different data structures which also reside in different areas of the memory.

How can we access these data structures since the variables we used with VDISYS do not work? ST Basic provides a system variable called GB which is a pointer



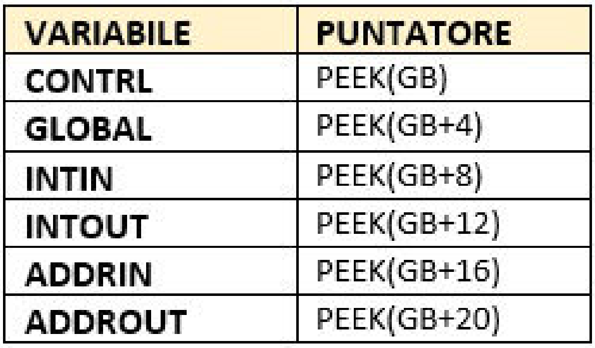
Returns the current mouse location, mouse button state and keyboard state.

BASIC CODE	DESCRIPTION
1 a#=gb	
2 gintout=peek(a#+12)	Define integer output
3 gemsys(79)	OPCODE
4 mx=peek(gintout+2)	mx=Coordinate of mouse's current location
5 my=peek(gintout+4)	my=Coordinate of mouse's current location
6 mb=peek(gintout+6)	mb=Current mouse button state. 0 No button pressed 1 Left button pressed 2 Right button pressed

BASIC CODE	DESCRIPTION
7 kb=peek(gintout+8)	kb= Current keyboard state 0 No key pressed 1 Right shift key pressed 2 Left shift key pressed 4 Control key pressed 8 Alternate key pressed

Fig. 4 - AES_OPCODE 73 - Mouse status

like any other, but it does not point to a data field, but to a table where other pointers are stored. And these are the pointers we need. To find the pointers to the structures we are interested in, we can use the following table:



VARIABILE	PUNTATORE
CONTRL	PEEK(GB)
GLOBAL	PEEK(GB+4)
INTIN	PEEK(GB+8)
INTOUT	PEEK(GB+12)
ADDRIN	PEEK(GB+16)
ADDROUT	PEEK(GB+20)

Unlike VDISYS, however, the GEMSYS function passes the OPCODE in the function call.

Have a look at the following programme:

```
10 b#=gb
15 I=peek(b+12)
20 cn=peek(b)
30 poke cn+8.0
40 poke cn+16.0
50 gemsys(79)
60 mx=peek(I+2)
62 mx=peek(I+4)
64 print "POS x= ";mx
65 print "POS y= ";my
```

To understand the programme, please refer to the information in Figure 4.

Warning: *I was really going crazy trying to figure out why, even though the presented listing was correct, I still got x=0 and y=252 as an answer.*
Then I found this article: http://gfabasic.net/stg/gfabasic.htm#GRAF_MKSTATE where, about the GRAF_MKSTATE function, it says "This is an AES routine to query the mouse. Unlike MOUSEX etc., the function gives valid results if the pointer is within a menu bar."

For a complete list of the functionalities offered by AES see: https://www.seasip.info/Gem/aes.html

I will stop here for the moment, well aware that I have only touched the top of the iceberg behind the ST Basic, the Atari ST and its TOS operating system.

Have fun!

Useful links

I report here links to some documents that I found useful in writing the article.

- https://www.atarimagazines.com/startv1n1/TrackingElusiveGDOS.html

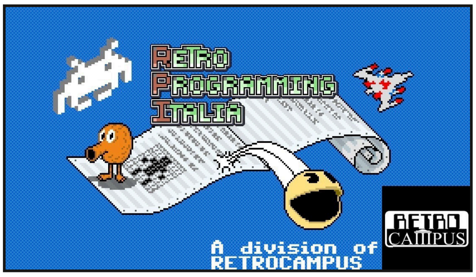
- http://www.atarimania.com/atari-magazine-atari-st-user_17.html

- https://docs.dev-docs.org/

- https://archive.org/details/Understanding_Atari_ST_BASIC_Programming/page/n165/mode/2up

How to create a game in BASIC for the Commodore 64 - part 2

by Felice Nardella (aka Kimono)

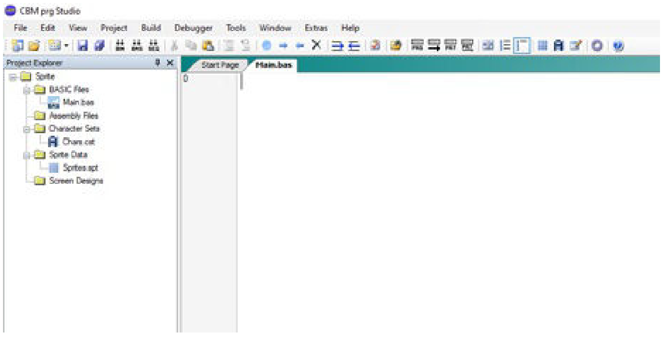


THE SPRITES (part #1)

Sprites is a topic of fundamental importance for videogames; it can be said that Sprites (besides the SID) have contributed a lot to the "fortune" that the C64 had in the '80s, and that still today, after almost 40 years(!), they give it a particular charm, thanks to which the remarkable interest that revolves around this extraordinary "vintage machine", despite its age, is still alive among us, incurable nostalgic people.

But let's proceed, now, without too many frills, with a brief description of the basic characteristics of the sprites: first of all let's say that, in Basic, the C64 is able to manage and display a maximum of 8 sprites at the same time, which are numbered from 0 to 7.

Well, to get familiar with sprites right away, start CBM prg Studio and create a new Basic Project (File → New BASIC Project) called "Sprite", taking care to leave the tick on "Add a Sprite Data file called" and rename the file "Main.bas" to "Balloon.bas".



Then click twice on "Sprites.spt" and the sprites creation screen will come up; after that go to "Sprites" → "New"; if you have done everything correctly, you will see a window like the one in figure 1.

As you can guess, we will have to draw our sprite in that grid and blacken its squares.

The grid consists of 21 rows and each row consists of 24 squares: each square represents 1 bit.

Since 8 bits make up 1 byte, each line consists of 3 bytes. In total, therefore, each sprite occupies 21 * 3 = 63 bytes. In reality, it takes up 64 bytes, because at the end of each sprite, there must be a null byte.

To summarise, each sprite occupies 64 bytes of RAM and is defined by 24 points in width and 21 points in height, for a total of 504 points (24 * 21).

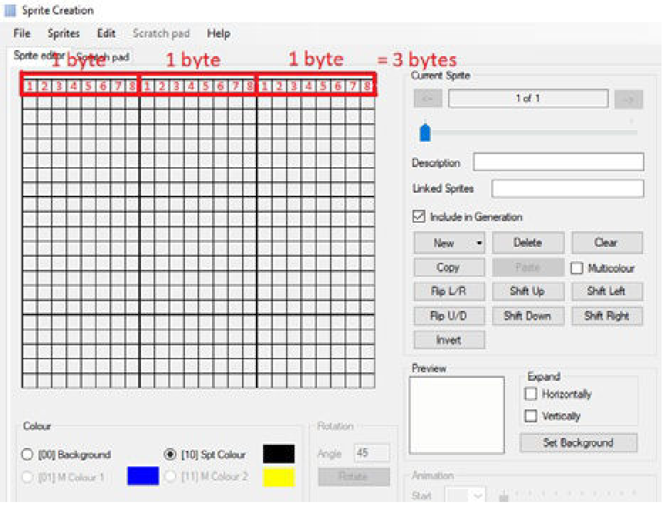
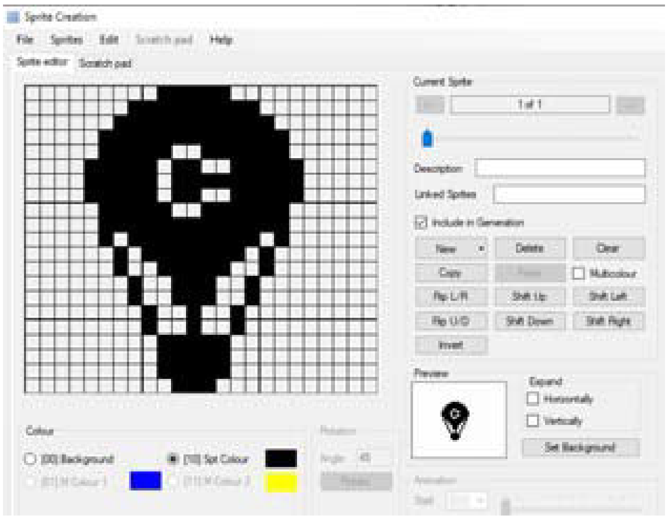


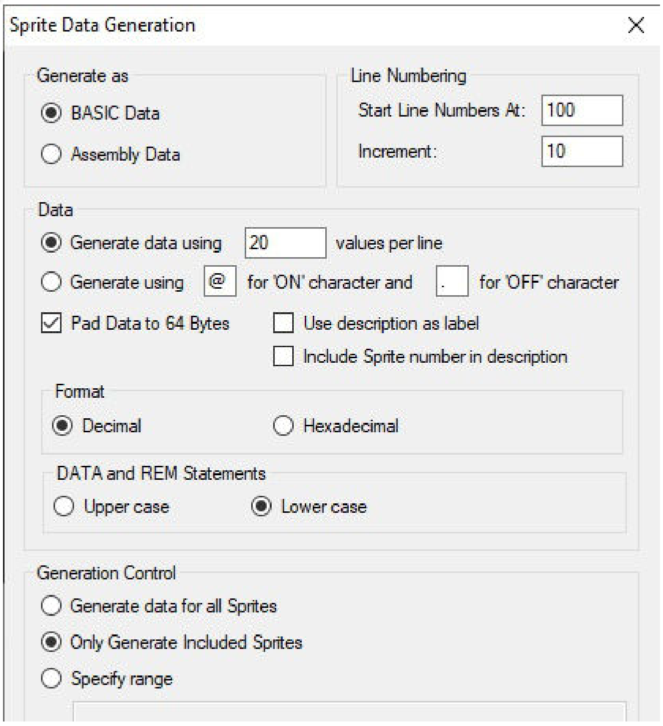
Fig. 1 - Sprite Editor within CBM Prg Studio

LET'S START!

Let's start drawing our sprite on the grid: who of you doesn't remember the famous hot air balloon presented in the classic manual that came with the C64? Well, now it's time to reproduce it on the CBM prg Studio! Try to draw a hot air balloon, like the one you see in the picture:



Then save it by clicking on "File" → "Save", then go to create the DATA of your sprite by clicking on "File" → "Export" → "To Listing..." and a window like this will open: Set everything as in the figure and press OK. The "CBM



Sprite Data Generation ×

Generate as
- ● BASIC Data
- ○ Assembly Data

Line Numbering
- Start Line Numbers At: 100
- Increment: 10

Data
- ● Generate data using 20 values per line
- ○ Generate using @ for 'ON' character and . for 'OFF' character
- ☑ Pad Data to 64 Bytes
- □ Use description as label
- □ Include Sprite number in description

Format
- ● Decimal
- ○ Hexadecimal

DATA and REM Statements
- ○ Upper case
- ● Lower case

Generation Control
- ○ Generate data for all Sprites
- ● Only Generate Included Sprites
- ○ Specify range

prg Studio" will now generate the following list with the DATA of the sprite:

```
100 rem
110 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
120 date 224,15,179,224,7,207,192,7,255,192,5,
255,64,2,254,128,2,124,128,1
130 given 57,0,1,57,0,0,146,0,0,146,0,0,124,0,
0,124,0,0,124,0
140 date 0,56,0,0
```

As you can probably guess, in the 1980s sprites were drawn by hand and the values of the drawing data, which CBM prg Studio calculated for us, were also calculated by hand!

Now that we have obtained the design data, thanks to CBM prg Studio, we must try to store them somewhere in RAM: in short, we must look for a space of at least 63 contiguous bytes; moreover, these blocks of bytes must start from locations that are multiples of 64.

To cut a long story short, for now you can take the following locations as a starting point for storing sprite data:

```
- 11 * 64 = 704
- 13 * 64 = 832
- 14 * 64 = 896
- 15 * 64 = 960
```

For example, we use the locations from 704 onwards by writing the following line of code:

```
10 for n=704 to 704+63: read q:poke n,q: next
```

POINTERS TO SPRITES

Once the sprite data has been stored, we need to let the C64 know where we have stored it, otherwise it will never be able to find it on its own.

To do this we will use the pointers to the sprites, which are in a block of 8 bytes starting from location 2040 to 2047. The location for the pointer to sprite 0 is 2040, the location for the pointer to sprite 1 is 2041, and so on for all 8 sprites.

Since we only have one sprite to display, we will currently only use the location 2040, which must refer to location 704.

SPRITE	PUNTATORI AGLI SPRITE
#0	2040
#1	2041
#2	2042
#3	2043
#4	2044
#5	2045
#6	2046
#7	2047

We will therefore have to add the following line to our programme:

```
20 poke 2040.11
```

This is because 704 / 64 = 11.

If, on the other hand, sprite 0 was, for example, stored from location 832, we would have to write

```
20 poke 2040.13
```

Since 832 / 64 = 13.

MAP OF REGISTERS

Now, in order to continue with our program, we will have to dwell for a moment on the chip that manages the graphics of the C64: the VICII.

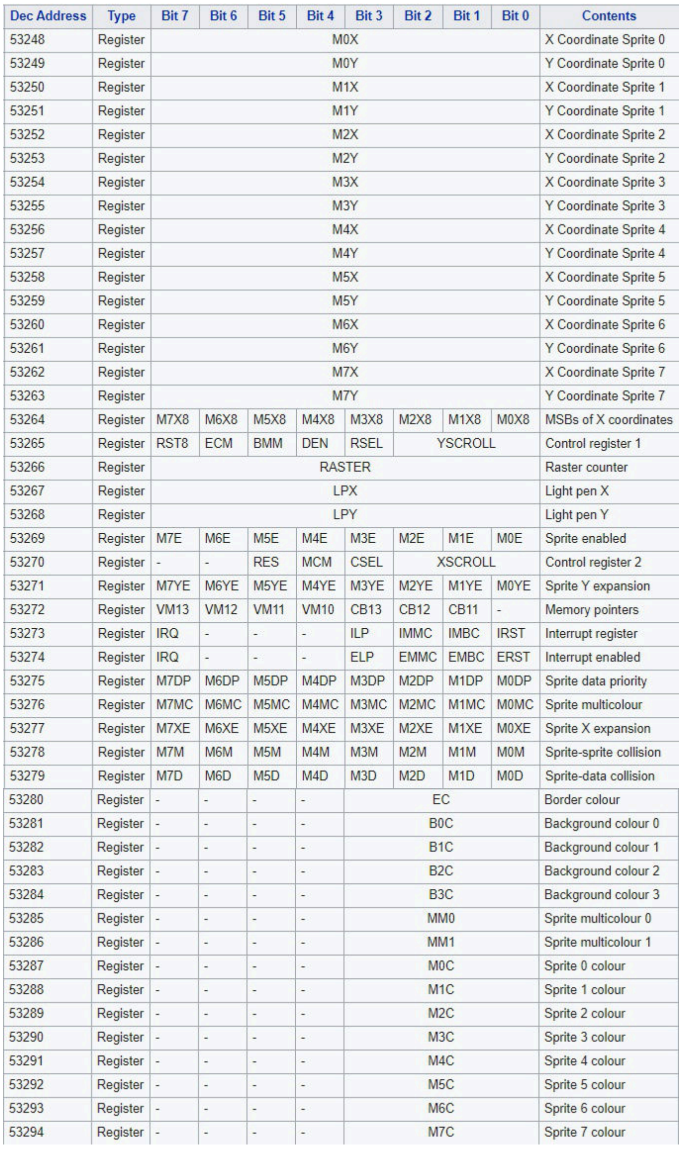
We will mainly need its registers, which start at byte 53248; refer to the VIC-II register map in Figure 2.

Let us analyse the registers which, for the moment, will serve our purpose of displaying a sprite:

a) Registers 53248, 53249 (Sprite Coordinates 0)

As you can see on the map, registers 53248 and 53249 define the x and y coordinates of sprite 0 respectively.

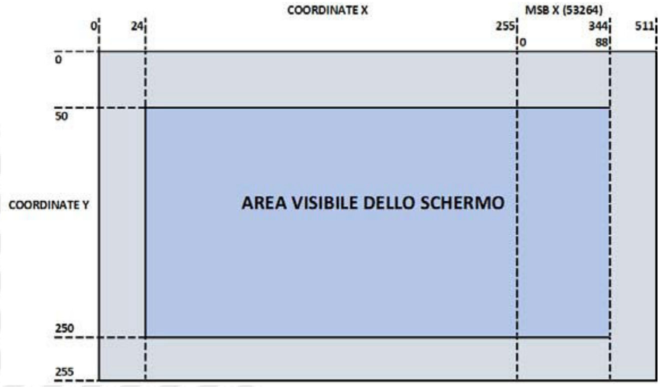
In order to display these coordinates on the screen, I have prepared the following figure which you can see below. As can be seen, the coordinates originate in the top left-hand corner. As for the Y co-ordinates, they range from 0 to 255, but those that are part of the visible area of the screen range from 50 to 250.



Dec Address	Type	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Contents
53248	Register	M0X								X Coordinate Sprite 0
53249	Register	M0Y								Y Coordinate Sprite 0
53250	Register	M1X								X Coordinate Sprite 1
53251	Register	M1Y								Y Coordinate Sprite 1
53252	Register	M2X								X Coordinate Sprite 2
53253	Register	M2Y								Y Coordinate Sprite 2
53254	Register	M3X								X Coordinate Sprite 3
53255	Register	M3Y								Y Coordinate Sprite 3
53256	Register	M4X								X Coordinate Sprite 4
53257	Register	M4Y								Y Coordinate Sprite 4
53258	Register	M5X								X Coordinate Sprite 5
53259	Register	M5Y								Y Coordinate Sprite 5
53260	Register	M6X								X Coordinate Sprite 6
53261	Register	M6Y								Y Coordinate Sprite 6
53262	Register	M7X								X Coordinate Sprite 7
53263	Register	M7Y								Y Coordinate Sprite 7
53264	Register	M7X8	M6X8	M5X8	M4X8	M3X8	M2X8	M1X8	M0X8	MSBs of X coordinates
53265	Register	RST8	ECM	BMM	DEN	RSEL	YSCROLL			Control register 1
53266	Register	RASTER								Raster counter
53267	Register	LPX								Light pen X
53268	Register	LPY								Light pen Y
53269	Register	M7E	M6E	M5E	M4E	M3E	M2E	M1E	M0E	Sprite enabled
53270	Register	-	-	RES	MCM	CSEL	XSCROLL			Control register 2
53271	Register	M7YE	M6YE	M5YE	M4YE	M3YE	M2YE	M1YE	M0YE	Sprite Y expansion
53272	Register	VM13	VM12	VM11	VM10	CB13	CB12	CB11	-	Memory pointers
53273	Register	IRQ	-	-	-	ILP	IMMC	IMBC	IRST	Interrupt register
53274	Register	IRQ	-	-	-	ELP	EMMC	EMBC	ERST	Interrupt enabled
53275	Register	M7DP	M6DP	M5DP	M4DP	M3DP	M2DP	M1DP	M0DP	Sprite data priority
53276	Register	M7MC	M6MC	M5MC	M4MC	M3MC	M2MC	M1MC	M0MC	Sprite multicolour
53277	Register	M7XE	M6XE	M5XE	M4XE	M3XE	M2XE	M1XE	M0XE	Sprite X expansion
53278	Register	M7M	M6M	M5M	M4M	M3M	M2M	M1M	M0M	Sprite-sprite collision
53279	Register	M7D	M6D	M5D	M4D	M3D	M2D	M1D	M0D	Sprite-data collision
53280	Register	-	-	-	-	EC				Border colour
53281	Register	-	-	-	-	B0C				Background colour 0
53282	Register	-	-	-	-	B1C				Background colour 1
53283	Register	-	-	-	-	B2C				Background colour 2
53284	Register	-	-	-	-	B3C				Background colour 3
53285	Register	-	-	-	-	MM0				Sprite multicolour 0
53286	Register	-	-	-	-	MM1				Sprite multicolour 1
53287	Register	-	-	-	-	M0C				Sprite 0 colour
53288	Register	-	-	-	-	M1C				Sprite 1 colour
53289	Register	-	-	-	-	M2C				Sprite 2 colour
53290	Register	-	-	-	-	M3C				Sprite 3 colour
53291	Register	-	-	-	-	M4C				Sprite 4 colour
53292	Register	-	-	-	-	M5C				Sprite 5 colour
53293	Register	-	-	-	-	M6C				Sprite 6 colour
53294	Register	-	-	-	-	M7C				Sprite 7 colour

Fig. 2 - VIC- II registers map

A few more words, however, must be spent on the X co-ordinates, which range from 0 to 511. As shown in the



figure, those that fall within the visible area of the screen start at 24 and go well beyond 255, but even so, it is not possible to go beyond the 255 co-ordinate unless you also act on register 53264, known as the Most Significant Bit register (or MSB).

This means that if we wanted to position sprite 0 at coordinate x = 256, we could not write POKE 53240,256 (no!), but would have to write:

```
POKE 53264,1 : POKE 53240.0
```

Since we want our hot air balloon to start from the bottom left-hand corner of the screen, we will add the following line:

40 poke 53248.24: poke 53249.229

Note that the value 229 is given by the max. coordinate of Y (250) from which the height of the sprite (21) has been subtracted.

In fact 250 - 21 = 229.

b) Register 53269 (enables sprites)

Another important location for our purposes is 53269, which is used to enable sprites, i.e. to make them visible, or, conversely, to disable them: for example, to enable sprite 0, we will need to add the following line to the program:

```
30 poke 53269.1
```

This is because 20 = 1.

If, on the other hand, we had to activate sprite 1, we would have had to write:

```
30 poke 53269.2
```

Since 21 = 2.

If yet, we had to activate both sprite 0 and sprite 1, we would have had to write:

```
30 poke 53269.3
```

Since 20 + 21 = 3

Similarly, in order to deactivate a sprite, the bit corresponding to it must be set to 0; for example, to turn off all the sprites (and thus make them disappear), it is necessary to write:

```
poke 53269.0
```

This will set all 8 bits to 0.

c) Register 53287 (sprite colour 0)

This register indicates the colour that our sprite 0 will have. In this case, we choose cyan, whose code, as we know, is 3, so we must also add the following line to the programme:

```
50 poke 53287.3
```

DISPLAYING THE SPRITE

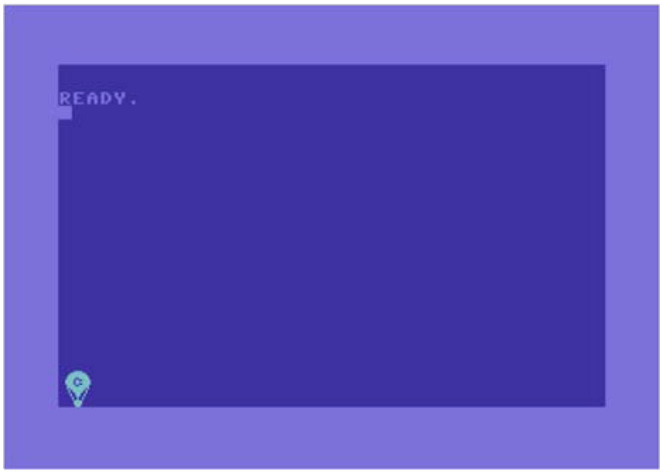
Finally, now that we've covered everything we need to create and display a sprite, we can type in our program, simply by summarising what we've described so far; then our little program will

will be presented as follows:

```
0 rem cleans the screen
1 print"{clear}"
5 rem loads the sprite data into memory
```

```
10 for n=704 to 704+63: read q:poke n,q: next
15 rem pointer to sprite 0
20 poke 2040.11
25 rem activates sprite 0
30 poke 53269.1
35 rem x, y coordinates of sprite 0
40 poke 53248.24: poke 53249.229
45 rem sprite colour 0 (3 = cyan)
50 poke 53287.3
100 rem sprite data 0
110 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
120 data 224,15,179,224,7,207,192,7,255,192,5,
255,64,2,254,128,2,124,128,1
130 data 57,0,1,57,0,0,146,0,0,146,0,0,124,0,
0,124,0,0,124,0
140 data 0,56,0,0
```

Now save it, compile it and run it on the emulator; this is what the screen will look like:



```
READY.
```

MOVE YOUR SPRITE!

Of course, having a static image is not the best way of life, so let's try to give it some movement! How do we do that? Simply by operating on the X,Y co-ordinates of the sprite. We want it to move from the bottom to the top and from the left to the right, so we need to increase the value of the X co-ordinate and simultaneously decrease that of the Y co-ordinate. One way of doing this is as follows:

```
65 rem increases x-coordinate and decreases
y-coordinate
70 for x=24 to 255 step 2: y=y-1
75 rem x, y coordinates of sprite 0
80 poke 53248,x: poke 53249,y
85 next
```

So the programme, in its entirety, will become:

```
5 rem cleans the screen
10 print"{clear}"
15 rem loads the sprite data into memory
20 for n=704 to 704+63: read q:poke n,q: next
25 rem pointer to sprite 0
30 poke 2040.11
35 rem activates sprite 0
40 poke 53269.1
45 rem sprite colour 0 (3 = cyan)
50 poke 53287.3
55 rem sets the initial y-coordinate
60 y=229
65 rem increases x-coordinate and decreases
y-coordinate
70 for x=24 to 255 step 2: y=y-1
75 rem x, y coordinates of sprite 0
80 poke 53248,x: poke 53249,y
85 next
90 rem repeats the cycle from line 60
95 goto 60
100 rem sprite data 0
105 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
110 data 224,15,179,224,7,207,192,7,255,192,5,
255,64,2,254,128,2,124,128,1
115 data 57,0,1,57,0,0,146,0,0,124,
0,0,124,0,0,124,0
120 data 0,56,0,0
```

Now we want the hot air balloon to overcome the "255 barrier" on the x-axis, so by using the MSB register, we can make our hot air balloon float along the entire length of the visible screen. This is what our programme will look like:

```
5 rem cleans the screen
10 print"{clear}"
15 rem loads the sprite data into memory
20 for n=704 to 704+63: read q:poke n,q: next
25 rem pointer to sprite 0
30 poke 2040.11
35 rem activates sprite 0
40 poke 53269.1
45 rem sprite colour 0 (3 = cyan)
50 poke 53287.3
55 rem sets the initial y-coordinate; clears
the bits in the MSB register
60 y=229:poke 53264.0
65 rem increases x-coordinate and decreases
```

```
y-coordinate
70 for x=24 to 255 step 2: y=y-1
75 rem x, y coordinates of sprite 0
80 poke 53248,x: poke 53249,y
85 next
90 rem enables the bit for sprite 0 in the
MSB register
95 poke 53264.1
100 rem increases x-coordinate and decreases
y-coordinate
105 for x=0 to 68 step2: y=y-1
110 rem x, y coordinates of sprite 0
115 poke 53248,x: poke 53249,y
120 next
125 rem repeats the cycle from line 60
130 goto 60
135 rem sprite data 0
140 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
145 data 224,15,179,224,7,207,192,7,255,192,
5,255,64,2,254,128,2,124,128,1
150  data  57,0,1,57,0,0,146,0,0,146,0,0,124,
0,0,124,0,0,124,0
155 data 0,56,0,0
```

DISPLAY MULTIPLE SPRITES

In order to be able to display several sprites at the same time, it is simply necessary to remember everything that has been described so far, always bearing in mind the register map. It is therefore necessary to:

1) Add as many pointers to the sprites (2040 to 2047) as there are sprites we want to display (max. 8);

2) Remember to activate them (register 53269);

3) Assign each one a position in terms of X,Y coordinates (registers 53248 to 53263);

4) Assign each one a colour (registers 53287 to 53294).

By following these 4 simple points, we will be able to represent several sprites together.

Of course, if we insert the same address (11 in our case) in the sprite pointers, we will display two or more identical sprites (perhaps of a different colour, but of the same shape). If, on the other hand, we wish to represent 2 or more different sprites, we will first have to insert the different shapes given to the sprites in RAM, through DATA, and then insert the addresses corresponding to the different sprites in the various pointers.

We will return to this aspect in the next chapter. For the moment, I will give a couple of examples of displaying several (identical) sprites at the same time: the first displays two hot air balloons swaying together, one at the top and the other at the bottom, with the help of the trigonometric functions sin() and cos().

Here is the complete commented list of the first example just described:

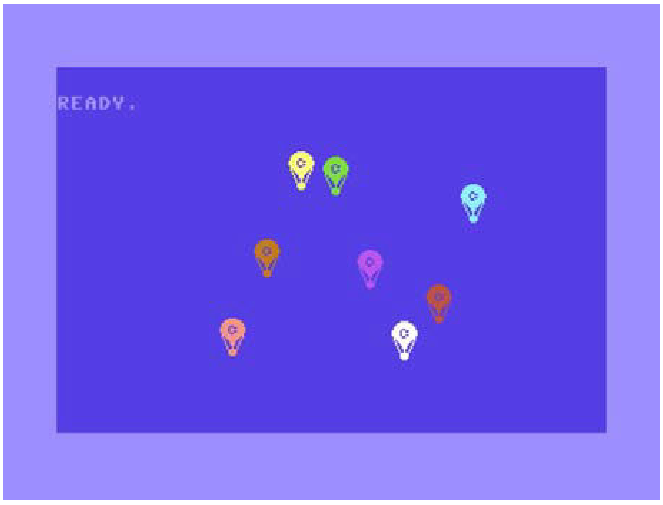
```
5 rem cleans the screen
10 print"{clear}"
15 rem loads the sprite data into memory
20 for n=704 to 704+63: read q:poke n,q: next
25 rem pointer to sprites 0 and 1
30 poke 2040,11:poke 2041,11
35 rem activates sprites 0 and 1
40 poke 53269.7
45 rem colour of sprites 0 and 1 (3 = cyan,
2 = red)
50 poke 53287.3: poke 53288.2
55 forx=24to255
60 rem calculates vertical trajectories
65 y=40*sin(x/2)+100:y1=40*cos(x/2)+180
70 rem x,y coordinates of sprites 0 and 1
75 poke 53248,x: poke 53249,y
80 poke 53250,x: poke 53251,y1
85 next
90 rem enables the bit for sprites 0 and 1
in the MSB register
95 poke 53264.3
100 rem x,y coordinates of sprites 0 and 1
105 forx=0to68
110 rem calculates vertical trajectories
115 y=40*sin(x/2)+100:y1=40*cos(x/2)+180
120 poke 53248,x: poke 53249,y
125 poke 53250,x: poke 53251,y1
130 next
135 rem sprite data 0
140 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
145 data 224,15,179,224,7,207,192,7,255,192,
5,255,64,2,254,128,2,124,128,1
150  data  57,0,1,57,0,0,146,0,0,146,0,0,124,
0,0,124,0,0,124,0
155 data 0,56,0,0
```

The next example shows no less than 8 different coloured hot air balloons, which sway in a row, one behind the other on the screen, tracing a sinusoidal trajectory.

Here is the complete list:

```
5 rem cleans the screen
10 print"{clear}"
15 rem loads the sprite data into memory
20 for n=704 to 704+63: read q:poke n,q: next
25 rem pointer to sprites
30 poke 2040,11:poke 2041,11:poke2042,11:poke
2043,11
35 poke 2044,11:poke 2045,11:poke2046,11:poke
2047,11
40 rem activates all 8 sprites
45 poke 53269,255
50 rem sprite colour
55 poke 53287,3:poke53288,2:poke 53289,1:poke
53290,4
60 poke 53291,5:poke 53292,7:poke53293,8:poke
53294,10
65 rem defines the function f(a)
70 def fn f(a)=50*sin(a)+140
75 rem x, y coordinates of sprites
80 forx=140to255
85 poke 53248,x: poke 53249,fn f(x)
90 poke 53250,x-20: poke 53251,fn f(x-20)
95 poke 53252,x-40: poke 53253,fn f(x-40)
100 poke 53254,x-60: poke 53255,fn f(x-60)
105 poke 53256,x-80: poke 53257,fn f(x-80)
110 poke 53258,x-100: poke 53259,fn f(x-100)
115 poke 53260,x-120: poke 53261,fn f(x-120)
120 poke 53262,x-140: poke 53263,fn f(x-140)
125 next
130 rem sprite data 0
135 data 0,124,0,1,255,0,3,255,128,7,255,192,
7,207,192,15,179,224,15,191
140 data 224,15,179,224,7,207,192,7,255,192,
5,255,64,2,254,128,2,124,128,1
145 data 57,0,1,57,0,0,146,0,0,124,
0,0,124,0,0,124,0
150 data 0,56,0,0
```

See you next issue with the third instalment of the tutorial!



```
READY.
```

Fig. 3 - The 8 ballons flying on the C64 screen

The entire editorial staff of RetroMagazine World thanks and appreciates the work of the Facebook group ***"RetroProgramming Italia - RP Italia" - A division of "RetroCampus" Associazione Culturale*** *and thanks them for their kind contributions.*

We believe that, like other similar groups, they are doing a great job to make known the Italian retrocoding scene in the world.

We invite our readers to collaborate with them in case they have material to share with all the fans of retrocomputing and retrocoding.

C128 in 80 column mode - part 3

by Gianluca Girelli

As mentioned in the previous parts, the 8563 VDC Chip is definetly a very powerful and versatile piece of equipment. We have learned that, unlike the VIC, its registers cannot be addressed directly using POKE/PEEKS commands, although said commands can be used to interact with the I/O registers. This lack of features makes interacting with the chip a little annoying, yet the results you can achieve with a little bit of tinkering are both outstanding and rewarding.

To give the reader a taste of what is actually possible, I decided to center the present article on a few short programs that show some of these inner possibilities. Hopefully those code snippets, taken form an old "Data Becker" book published by "Abacus Software" in 1985, will make you craving for further and deeper testing.

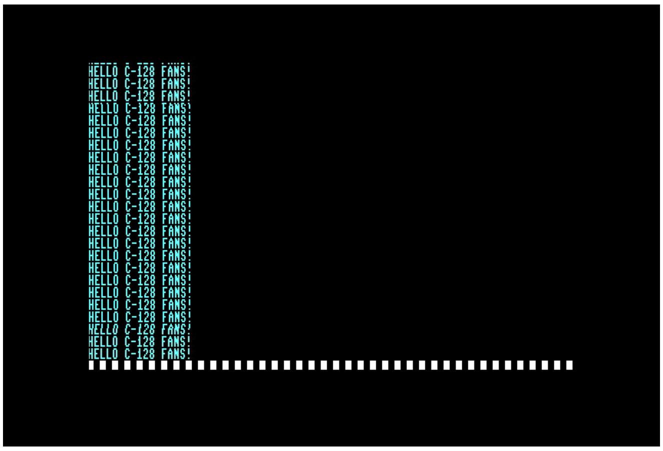
Without further ado, let's start with a program that shows how smooth scrolling of the video can be achieved. As with the VIC chip, you can move the screen vertically or horizontally in raster line increments on the VDC; this is done by using the VDC register 24 and 25 ("VERTICAL SMOOTH SCROLL" and "HORIZONTAL SMOOTH SCROLLING" rexpectively) where its 0-4 (for register 24) and 0-3 bits (for register 25) are used for this purpose. Contrary to the way smooth scrolling is done on the VIC, you don't lose any columns or lines on the VDC. The VDC is not well-suited for games: although it has very good resolution, its complicated addressing is far too slow to be useful in a game, yet you can use smooth scrolling to create many useful effects.

If this goes too fast for you or not fast enough, change the delay loops in lines 130 and 160 correspondingly.
If bit 3 is cleared, 25 lines are displayed and the following (or preceding) RAM is scrolled on the screen (remember from previous articles that the VDC has its own indipendent memory). If you set bit 3, only 22 lines are displayed and you can scroll the last three lines of the screen by means of smooth scrolling.

```
10 rem *** demo program for smooth scrolling ***
20 a=dec("d600"):d=dec("d601")
30 ve=24:ho=25
40 print chr$(147)chr$(27);"m";: rem screen clr
and scroll off
50 a$="hello c-128 fans!"
60 for i=0 to 24
70 print a$
80 next
90 :
100 for i0=0 to 6
110 poke a,ve:v=peek(d) and 240 or i0
120 poke a,ve: poke d,v
130 for i1=1 to 20: next
140 poke a,ho: h=peek(d) and 240 or i0
150 poke a,ho: poke d,h
160 for i1=1 to 20: next
170 next
180 goto 100
```

As you can see from Figure 1, the screen shows corrupted lines at its bottom; we will get to that in a bit. For the time being, let's focus on another point: we just said that this feature is not fast enough for games nonetheless, if the controller is so hard to access, why is screen scrolling so fast? The solution is simple: The VDC is intelligent enough to move entire blocks in its memory. If this had to be done via the relative addressing, it would take a considerably longer time.

If you want the VDC to move an area of memory, you must tell it this via the COPY bit (bit 7 in REG 24). If this bit is



```
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
HELLO C-128 FANS!
▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮
```

Fig. 1

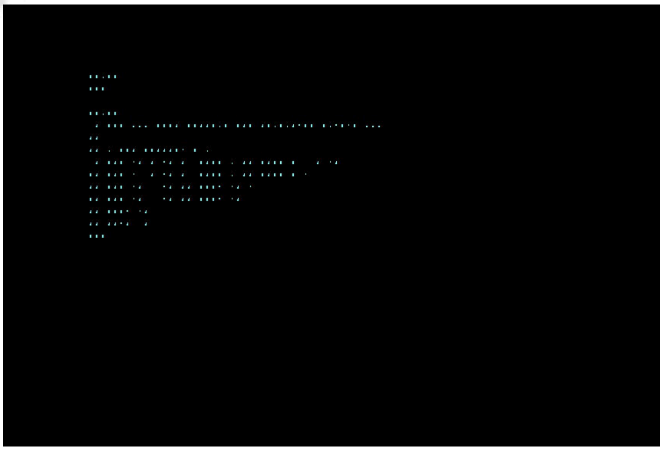


Fig. 2

set, the VDC copies instead of filling. The starting address of the block to be copied is defined in registers 32 and 33; the destination address of the copying procedure must be defined in the update register (REG 18 and 19); the copy process begins when you write to the word count register. This also specifies the number of characters to be copied.

NOTE: *The word count register specifies the exact number of characters to be copied. For example, if you want to copy the first text line on the screen to the line below and preserve the attributes, you must first copy the text line and then the attributes. We will do an upward-scroll in our example program; in BASIC it goes quite slowly, but in machine language it is fast enough.*

In this particular case a picture wouldn't help but, if you run the code, you will notice the entire screen moving upward by means of block-copying.

```
10 rem *** demo program for block copying ***
20 a=dec("d600"):d=dec("d601")
30 poke a,24: c=peek(d):rem *** contents of reg 24
40 poke a,24: poke d,c or 128:rem *** set copy bit
50 for z=24 to 0 step -1
60 aq=z*80: az=aq+80: rem * * * source and dest
70 poke a,18:poke d,az/256:poke a,19:poke d,az and 255
80 poke a,32:poke d,aq/256:poke a,33:poke d,aq and 255
90 poke a,30:poke d,79: rem *** copy text
100 aq=2048+aq:az=2048+az:rem *** attribute addr
110 poke a,18:poke d,az/256:poke a,19: poke d,az and 255
120 poke a,32:poke d,aq/256:poke a,33: poke d,aq and 255
```

Fig. 3

```
130 poke a,30: poke d,79: rem *** copy attribute
140 next
150 print chr$(19);chr$(27)"d"; :rem clear 1st line
160 poke a,24:poke d,c: rem *** clear copy bit
```

Let's now explore how the characters' matrix can be altered. As you guys know, similar to the VIC chip, even the matrix of the characters found in VDC RAM is 8x8 points; this means that the characters displayed on the screen are 8 points wide and 8 lines tall. This can be changed by tinkering the registers 22 and 23, "CHARACTER TOTAL & DISPLAYED" and"CHARACTER DSP(V)" respectively, where the first one determines the total number of displayed horizontal lines (and the width of the character) while the second determines the number of vertical lines (and the height of a character). Figures 2 to 4 shows the code while being executed.

```
10 rem *** demo program for character matrix ***
20 :
30 a=dec("d600"):d=a+1
40 for i0=0 to 8: poke a,22:poke d,112+i0
50 for i1=0 to 8: poke a,23:poke d,i1
60 for i2=1 to 30:next i2,i1
70 for i2=1 to 30:next i2
80 next i0
90 goto 40
```

It's now time to delve into another very remarkable feature of the VDC: the ability to display more then 25 rows on screen: such feature, coupled with the 80 columns, adds even more flexibility to an already out-of-the-ordinary computer (for 1985, that is). This characteristic was well suited for all the programmers who wanted to code word processors or database applications and expanded the 80x25 matrix (2000 characters on screen) to an 80x28 matrix (2240 characters); definetly a huge upgrade if compared to the 40x25 standard of the time.

Let's start with some theory:
In register 6 of the video controller ("VERTICAL DISPLAYED"), you can specify how many lines are to appear on the screen. The default value here is 25. Let's change this value to 10:

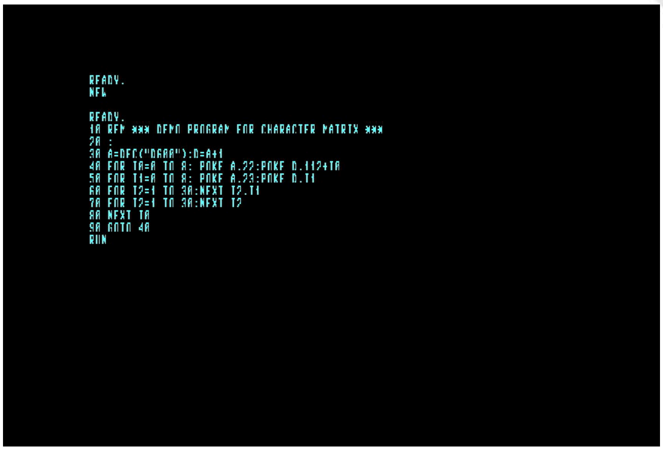
```
10 A=DEC("D600"):D=A+1
20 POKE A,6:POKE D,10
```

If you run the code you'll see that the controller now displays only 10 lines on the screen and the remaining lines are simply "swallowed up." Just as we can make the screen smaller, we also have the ability to increase the number of lines. We do this by simply correcting line 20:

```
20 POKE A,6: POKE D,28
```

And now we have 28 lines on the screen. You also see some lines that will usually flash in various colors. We can now (provided the monitor is good enough) see all 28 lines on the screen, even if the last three lines don't contain any useful information.

We already know that the video RAM lies at address $0000 and the attribute RAM at address $0800.
We must change this since we have displayed 2240 characters; the end of the video RAM then lies at address $0960 and part of the attribute RAM is overwritten (and vice versa). There is enough space between the attribute RAM and the character generator. Address $0A00 is then available for the start address of the attribute RAM. But when we want to write to the 80-column screen with BASIC, we have a small problem:
the interpreter gets the base address of the attribute RAM from address $0A2F in the zero page. This isn't so bad —we just inform the BASIC interpreter of the new base address. Although logically correct, this is not how it works since if we take a closer look at the kernal, we see that the base address is not added but logically ORed.
Bits 0 and 1 are affected by this; these two bits may not be relevant; that is, they may not be set.
This is why it is advisable to define address $1000 as the start address of the video RAM. We do this with the two instructions:



```
READY.
NEW

READY.
10 REM *** DEMO PROGRAM FOR CHARACTER MATRIX ***
20 :
30 A=DEC("D600"):D=A+1
40 FOR T0=0 TO 8: POKE A,22:POKE D,112+T0
50 FOR T1=0 TO 8: POKE A,23:POKE D,T1
60 FOR T2=1 TO 30:NEXT T2,T1
70 FOR T2=1 TO 30:NEXT T2
80 NEXT T0
90 GOTO 40
RUN
```

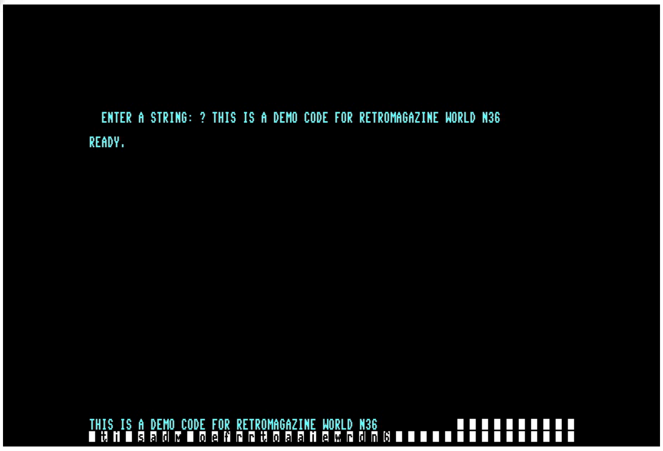
Fig. 4

```
POKE DEC("0A2F"),16
POKE DEC("D600"),20:POKE DEC("D601"),16
```

At this point everything should be set yet: when you start this program 28 lines appear on the screen but the last three lines still have no meaningful content. Unfortunately, we cannot write to these lines with the PRINT statement: the operating system is not prepared for such things. It becomes clear that we must POKE characters (strings) into memory. This is done by a small machine language routine so that the characters to be printed can be put into a string.

Analizing the machine language code for this routine is beyond the scope of this article so, for the time being, we can make good use of the following loader. Simply run it to load the LM code into memory and save the resulting binary code for later use: line 120 will help you in the process.

```
10 rem basic loader for print string
20 :
30 for i=dec("d00") to dec("d79")
40 read a$
50 poke i,dec(a$)
60 s=s+dec(a$)
70 next
80 if s<>16613 then print"error in data statements"
90 input"save program on diskette y/n";a$
100 if a$<>"y" then end
110 input"file name";f$
120 bsave""+f$+"",b1,p3328 to p3449:end
130 :
200 data 8e,00,d6,2c,00,d6,10,fb,8d,
01,d6,60,a2,12,a9,00
210 data
20,00,0d,e8,a9,00,20,00,0d,a2,1f,a9,00,20,00,0d
```



```
  ENTER A STRING: ? THIS IS A DEMO CODE FOR RETROMAGAZINE WORLD N36
READY.

THIS IS A DEMO CODE FOR RETROMAGAZINE WORLD N36
```

Fig. 5

```
220 data a2,12,a9,00,4c,00,0d,85,fc,
86,fd,a0,00,a2,01,a9
230 data fc,20,74,ff,85,fe,a0,01,a2,01,a9,fc,
20,74,ff,48
240 data c8,a2,01,a9,fc,20,74,ff,85,fd,
68,85,fc,a5,fc,d0
250 data
02,c6,fd,c6,fc,a5,fa,d0,02,c6,fb,c6,fa,a5,fa,85
260 data e0,a5,fb,85,e1,a2,01,a4,fe,a9,fc,
20,74,ff,a4,fe
270 data 84,ec,20,0c,c0,c6,fe,d0,e4,60
```

With this step done, run the following code to finally access a full-fledged 80x28 hi-res screen where the last 3 bottom lines can be used as status lines inside your programs: the possibility are endless! Figure 5 shows one of them.

Basically what the code does is:
first a string variable is defined which contains the string to be printed. Then we POKE the start address in $FA and $FB, low byte first. We then indicate the address at which the string T$ is stored in bank 1 by using the BASIC command POINTER(variable name).

This address is then divided into low and high bytes and passed to the output routine at address $0D27.
The routine then gets each character and outputs it. That's it.

```
10 rem *** demo program for 28 line screen ***
20 rem load and run previous program first
30 a=dec("d600"):d=dec("d601")
40 poke a,20:poke d,16:rem *** vdc gets new base
50 poke dec("0a2f"),16:rem *** kernal gets new
base address
60 poke a,7:poke d,28:rem *** 28 lines
70 poke a,6:poke d,33:rem *** new sync
80 :
90 print chr$(147);
100 t$="                    ":rem 20 spaces
110 for x=0 to 79 step 20:for y=0 to 2
120 gosub 1000:next:next
130 input"enter your name: ";t$
140 for y=0 to 2:x=2*y:gosub 1000:next
150 end
1000 rem *** output t$ at x,y coordinate; y=0
means 1st line ***
1010 az=2000+y*80+x:rem destination address
1020 poke 250,az and 255:rem low byte
1030 poke 251,az/256:rem high byte
1040 t%=pointer(t$):rem address of the string
1050 sys dec("d27 "),t% and 255,t%/256: rem pass
1060 return
1070 :
```

Once again, this brief article shows how actually advanced was a computer which, at the time, suffered from the (well deserved) reputation of the C64 and the looming shadow of the upcoming Amiga range of computers.

For a true 8-bit coder, though, coding the C128 is Heaven on Hearth, since it still proves to be an incredible "educational" machine where to learn and test your skills.

Stay tuned!

Bibliography:

Commodore 128 Internals, an authoritative insider's guide
By K.Gerits, J.Schieb & F.Thrun
A Data Becker Book Published by Abacus Software, 1985-1986

BASIC fractals in 8-bit sauce - part 4

by David La Monaca

Despite of what I thought when I decided to start this little column on RetroMagazine World, we are now at the fourth episode and our beloved 8-bit computers, even with their graphical limitations, never cease to amaze us. Other contributions have been received from many readers, both original works for new 2D fractals, and conversion of listings already published for more or less known BASIC languages and for more or less famous machines. In short, our goal of arousing new interest in 8-bit BASIC programming, through the attraction of high-resolution graphics and vivid colours, has fully succeeded. And if we continue to receive interesting proposals and listings from our readers, you can be sure that we will also reach the fifth round of this column.

In this issue, the BASIC of the Sinclair ZX Spectrum gets most of the space, with listings being conversions of fractals appeared in the previous issues or implementations optimized for this machine, since they exploit some particular memory locations or are written in order to seek the maximum speed in plotting the graph or the brevity of the listing. Many of the listings are accompanied by comments and brief descriptions to facilitate the understanding of the code, so you can even modify it in order to obtain some additional effect or simply more colour. We are very thankful to the many authors of original listings or conversions from other 8-bit BASICs, all of whom are mentioned before the code. So without further ado let's begin the review of fractals selected for this column.

20. ZX Spectrum BASIC – Spiral of Ulam

by **Matthew Logue**, from the FB group "BASIC on ZX Spectrum"

```
10 PLOT INVERSE 1;128,88: LET p=1: FOR a=1
TO 176: LET b=-a*(COS (a*PI)): LET x2=PEEK
23677: LET y2=PEEK 23678:
20 FOR x=(SGN b) TO b STEP SGN b: LET x1=x2+x:
LET p=p+1: FOR f=2 TO SQR p: IF p=f*INT (p/
f) THEN PLOT INVERSE 1;x1,y2: NEXT x: GO TO 35:
30 NEXT f: PLOT x1,y2: NEXT x
35 LET x2=PEEK 23677: LET y2=PEEK 23678:
FOR y=(SGN b) TO b STEP SGN b: LET p=p+1:
LET y1=y2+y: FOR f=2 TO SQR p: IF p=f*INT
(p/f) THEN PLOT INVERSE 1;x2,y1: NEXT y: NEXT a
40 NEXT f: PLOT x2,y1: NEXT y
50 NEXT a
```

21. ZX Spectrum BASIC – Spiral of Ulam and much more...

by **Simon Férré**, FB group "BASIC on the ZX Spectrum"

The program can draw the Ulam spiral and other spirals by plotting only one point on the screen every N computed. Changing the value of the variable c in line 10 will change the "step" to draw a normal spiral. By changing the value of pr in line 10, the program will draw either the Ulam spiral (pr=1) or a normal one (pr=0). A value of 2 for variable c will produce a chequered pattern, while a value of 8 will generate two different patterns on either side of an imaginary diagonal starting from the origin (the centre of the screen).

```
10 LET c=7: LET x=127: LET y=88: LET md=1:
LET n=1: LET p=1: LET pr=1
20 FOR h=1 TO ABS md
30 IF pr=0 AND p=c THEN IF x>=0 AND x<=256
AND y>=0 AND y<=175 THEN PLOT x,y
40 IF pr=1 THEN IF x>=0 AND x<=256 AND y>=0
AND y<=175 THEN GO SUB 140: IF NOT d THEN
PLOT x,y
50 LET n=n+1: LET p=p+1-(p AND p=c): LET
x=x+(1 AND md>0)-(1 AND md<0): NEXT h
60 IF n>=65536 THEN GO TO 130
70 FOR v=1 TO ABS md
80 IF pr=0 AND p=c THEN IF x>=0 AND x<=256
AND y>=0 AND y<=175 THEN PLOT x,y
90 IF pr=1 THEN IF x>=0 AND x<=256 AND y>=0
AND y<=175 THEN GO SUB 140: IF NOT d THEN
PLOT x,y
100 LET n=n+1: LET p=p+1-(p AND p=c): LET
y=y+(1 AND md>0)-(1 AND md<0): NEXT v: LET
md=-1*(md+(1 AND md>0)-(1 AND md<0))
110 IF n>=65536 THEN GO TO 130
120 GO TO 20
130 STOP
140 LET d=0: IF n=1 THEN LET d=1: RETURN
145 IF n=2 OR n=3 THEN RETURN
150 IF n/2=INT (n/2) THEN LET d=1: RETURN
155 IF n<17 THEN FOR i=3 TO n-1 STEP 2: LET
d=d+(n/i=INT (n/i)): GO TO 165
160 FOR i=3 TO SQR (n) STEP 2: LET d=d+(n/
i=INT (n/i))
165 IF d=0 THEN NEXT i
170 RETURN
```

22. ZX Spectrum/ZX81 Basic – Spiral

by **Pierluigi Pieroli**, from the FB group Z80

```
10 INPUT N: PRINT AT 1,6;N
20 LET D=2*PI/600
30 LET P=84
35 PLOT 128,88
40 FOR I=0 TO 600
50 LET T=D*I
55 LET R=P*SIN (N*T)
60 LET X=R*COS T+128
70 LET Y=R*SIN T+88
80 DRAW X-PEEK 23677,Y-PEEK 23678
90 NEXT I
```

Replace/add the following lines to get a bit of colour:

```
5 PAPER 0: BORDER 0: CLS
80 INK x/32+1: DRAW x-PEEK 23677, y-PEEK 23678
```

23a. QuickBasic 3.5 PC - Archimedes' spiral

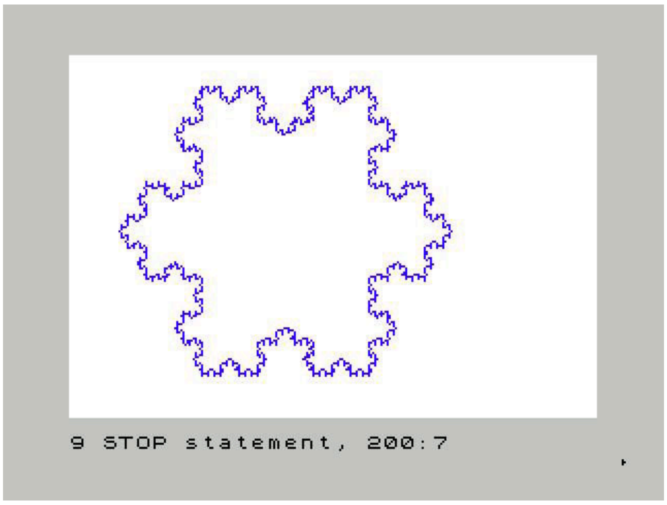
```
10 REM * ARCHIMEDES SPIRAL *
30 SCREEN 11 : CLS : PI=3.141592654
40 WINDOW (-12,-9)-(12,9)
50 A=.1 : PSET(0,0)
60 FOR T=0 TO 16*PI STEP .1 : R=A*T
70 X=R*COS(T)  : Y=R*SIN(T)
80 LINE -(X,Y)
90 NEXT T
100 A$=INPUT$(1)  : END
```

23b. QuickBasic 3.5 PC - Logarithmic spiral

```
10 SCREEN 11: CLS : PI = 3.141593
20 WINDOW (-4, -3)-(4, 3)
30 A = .05: B = .1 'starting point & growth rate
40 PSET (A, 0)
50 FOR T = 0 TO 42 STEP .1: R = A * EXP(B * T)
60 X = R * COS(T): Y = R * SIN(T)
70 LINE -(X, Y)
80 NEXT T: A$ = INPUT$(1): END
```

24. ZX BASIC – Sierpinsky Triangle

```
10 PAPER 0: BORDER 0: BRIGHT 1: CLS
20 LET x=160*RND: LET y=140*RND
25 LET ox=47: LET oy=157: REM xy offsets
30 LET c=INT(3*RND)
40 LET x=x/2: LET y=y/2
50 IF c=0 THEN LET x=x+40: LET y=y+70: GO TO 70
60 IF c=1 THEN LET x=x+80
70 PLOT INK 6-c;ox+x,oy-y
80 GO TO 30
```



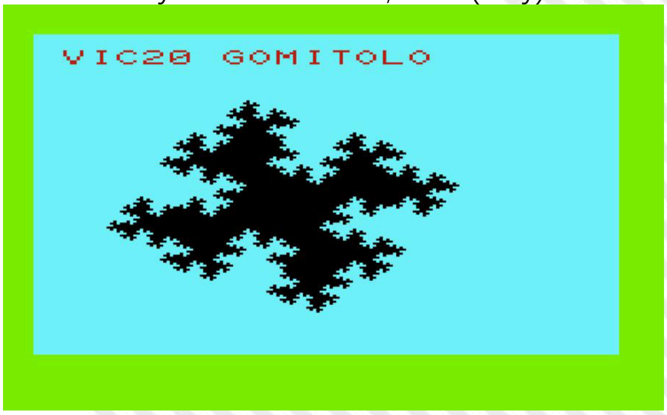
```
9 STOP statement, 200:7
```

25. ZX SPECTRUM BASIC (Z80, 1984) – Koch snowflake

```
10 LET l$="F--F--F"
20 FOR i=1 TO 3
30 LET n$=""
40 FOR j=1 TO LEN l$
50  IF  l$(j)="F"  THEN  LET  n$=n$+"F+F--F+F":
GO TO 70
60 LET n$=n$+l$(j)
70 NEXT j
80 LET l$=n$
90 NEXT i
100 LET ang=0
110 PLOT 174,5
120 FOR i=1 TO LEN l$
130 IF l$(i)="F" THEN DRAW 6*SIN ang,6*COS ang
140 IF l$(i)="+" THEN LET ang=ang+PI/3
150 IF l$(i)="-" THEN LET ang=ang-PI/3
160 NEXT i
```

26. VIC-20 + SUPER EXPANDER JP (6502, 1981) - Woolball

Conversion by **Armando Pavese**, Biella (Italy)

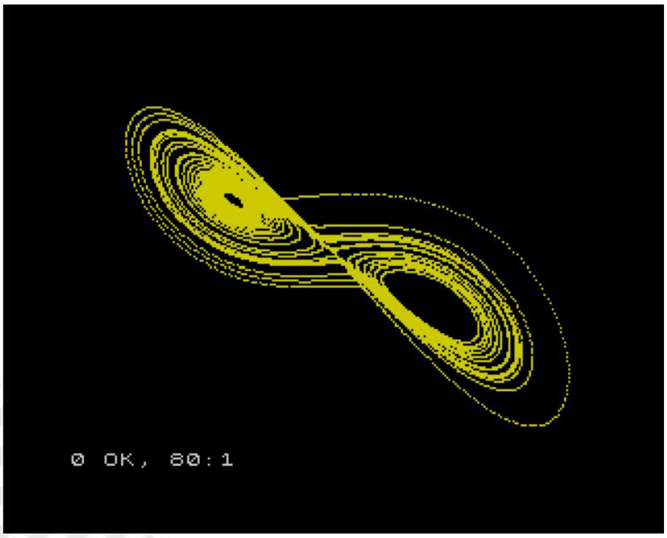


```
VIC20 GOMITOLO
```

```
10 hires0
15 setc1,4,6
110 dimx(16):dimy(16)
120 fori=1to4
130 forj=1to4
140 k=4*i+j-4
150 x(k)=j-3:y(k)=i-3
160 nextj,i
170 x(2)=0:y(2)=-3
180 x(8)=2:y(8)=0
190 x(9)=-3:y(9)=-1
200 x(15)=-1:y(15)=2
210 fori=1to16
220 forj=1to16
230 fork=1to16
240 xx=16*x(i)+4*x(j)+x(k)
250 yy=16*y(i)+4*y(j)+y(k)
260 plotxx+85,-yy+75
270 nextk,j,i
280 setc3:char1,1,"vic20 gomitolo"
300 wait198,1:poke198,0
310 text:setc7,2,4:end
```

27. ZX Basic - Lorenz Attractor

Conversion from Dragon BASIC by **Anja de Weerd** (Eindhoven, The Netherlands)



```
0 OK, 80:1
```

```
10 REM PROGRAM Lorenz system *
20 BORDER 0: PAPER 0: INK 6: CLS
30 LET rho=10: LET sigma=28: LET beta=8/3:
REM Constants for this type of Lorenz system
35 LET d=.002: REM Plotting density/distance
40 LET x=.1: LET y=0: LET z=0: REM Starting
values for the coordinates
50 FOR i=1 TO 16000
60 PLOT 127+5.7*x,87-3*y
70 LET x=x+d*rho*(y-x): LET y=y+d*(x*(sigma-
z)-y): LET z=z+d*(x*y-beta*z): REM Algorithm
to change the coordinates
80 NEXT i
```

28. ZX Basic – Archimede's spiral

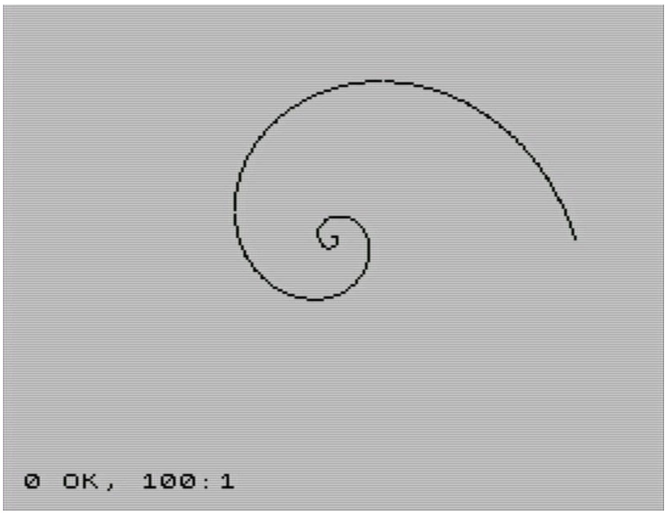
by **Uwe Geiken**, Germany

The factor a determines how close the lines are. The variable r is the linearly increasing radius, p is the angle and s is the step for the angle. After a complete circle, a new step is calculated so that the points are close together (line 35).

```
10 LET a=.6: LET r=1: LET p=0: LET s=.2
20 LET x=128+r*COS p: IF x>=0 THEN IF x<=255
THEN LET y=88+r*SIN p: IF y>=0 THEN IF y<=175
THEN PLOT x,y
30 LET p=p+s: LET r=r+a*s
35 IF p>2*PI THEN LET p=p-2*PI: LET s=.8/r
40 GO TO 20
```

29. ZX Basic – Fibonacci Spiral

by **Richard Scaglione**, FB group "BASIC on ZX Spectrum"

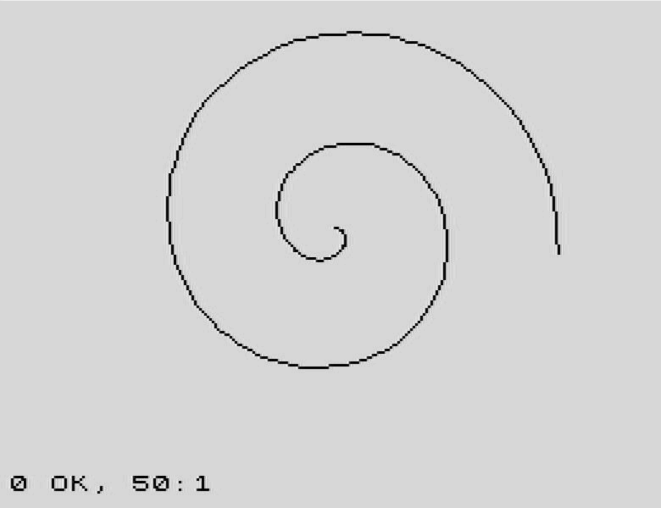


```
0 OK, 100:1
```

```
10 LET p=2*COS (PI/5)
20 LET c=0
30 LET r=100
40 LET x=127+r*COS c
50 LET y=87+r*SIN c
60 PLOT x,y
70 LET d=ASN (1/r)
80 LET r=r*((1/p)^(d/(PI/2)))
90 LET c=c+d
100 IF INT r THEN GOTO 40
```

30. ZX Basic – Fibonacci's spiral

by **Slavo Labsky** – optimized for speed



```
0 OK, 50:1
```

This makes use of locations 23677 and 23678 of the Spectrum to retrieve the latest plotted point.

```
10 PLOT 216,78
20 FOR r=88 TO 0 STEP -.5
30 LET u=r*.144
40 DRAW 128+r*COS u-PEEK 23677,88-r*SIN u-
PEEK 23678
50 NEXT r
```

31.ZX Basic – Golden spiral

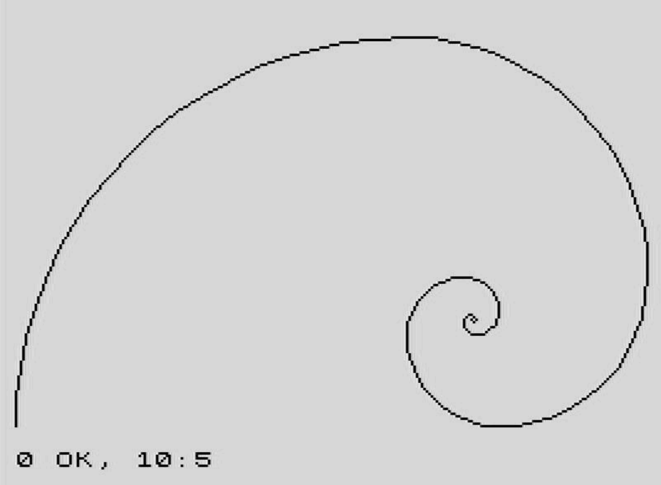
by **Uwe Geiken**, Germany

If you change the STEP to a value of 1, you will see that the points have all the same distance.

```
54 FOR r=1 TO 189 STEP .2: LET p=LN (r*4.175)/.
3: PLOT 183+r*COS p,45+r*SIN p: NEXT r
```

32. ZX Basic – Golden spiral 1

by **Matthew Logue**, UK – version using DRAW (120 bytes)



```
0 OK, 10:5
```

```
10 LET a=255: LET b=1.618: FOR x=1 TO 12:
DRAW (((INT ((x+1)/2)-2*INT ((x+1)/4))*2)-
1)*(a/b^x),(-(((INT ((x)/2)-2*INT ((x)/
4))*2)-1)*(a/b^x)),-PI/2: NEXT x
```

33. ZX Basic – Golden spiral 2

by **Matthew Logue**, UK – version using PLOT (57 bytes)

```
10 FOR a=1 TO 19.1 STEP .001: LET r=-.615*EXP
(.3*a): PLOT r*COS a+183,r*SIN a+45: NEXT a
```

With this article, the column “BASIC Fractals in 8-bit sauce”, which was started partly for fun and partly to stimulate our readers to put their hands back on the keyboard of their 8-bit machines, has reached its end (and as said, also its goal). In the future, we may resume the publication if we receive more input from readers and fans of graphics generated from mathematical formulas. This is, after all, an enormous territory to explore.

As usual, we invite you to send your ideas, thoughts and listings, best if accompanied by a few lines of information and/or instructions, to the e-mail:
RetroMagazine.Redazione@gmail.com

Any entry will get a reply and, if we reach an interesting critical mass, we'll organise a new episode of "Fractals in BASIC in 8-bit sauce" in the next issues of the magazine. For now, thanks again to all those who have contributed or participated in the success of this series and see you soon with more intriguing topics related to 8-bit graphics!

Interview with Andrea Contato

by Alessandro Albano

I met the protagonist of this interview by chance, after having replied to a post of his on Facebook. He was talking about his book on videogames, claiming that it was very different from the usual in the genre. I was curious, so I contacted him and started chatting with him on Messenger. I was very impressed by his passion and availability.

Then I started to read his book, or rather, I started to listen to it as I am blind. Now I'll reveal his name: Andrea Contato was immediately sensitive to the issue of accessibility and I must say that his book in ePub format was immediately "without barriers", making me passionate too, more impressed by the contents than by the many images. So I told myself that many people could really enjoy this production and I decided to interview him. I liked "Stories of video games...", I read it in one go and can't wait to read the sequel. And now we leave the word to Andrea.

Hi Andrea, would you like to tell us a bit about yourself?

I am a video gamer. I've always been one, and presumably I will be in the future. Nowadays I have less time to play games, partly because I do a lot of research and writing, but when the games I often talk about were just released and were state of the art, I was young and had plenty of free time to try them all.

I was born in 1976, so I missed the very first generation of video games. I've only seen it in deferred form, in the form of old, half-disused booths in village bars. But the first big wave of consoles and computer games, at least here in Italy, came when I was old enough to be able to read and write and enjoy them all.

I have always had three great passions: video games, history and reading/writing.

At some point I found the right balance and combined these three interests into one activity: being a researcher, a video game historian and writing books on the history of video games.

Tell us about it.

My first experiment was Through the Moongate: the story of Richard Garriott, Origin Systems Inc and Ultima. The venture arose from the realisation that there was no monograph on Garriott, his most important company and the computer role-playing game series (one of the longest and most important in history). Yes, there was an old work written by Shay Addams, but it was a work commissioned directly by Origin and Garriott himself, interesting, well written, but certainly not objective.

I decided to fill this gap myself, and as I set to work, a bigger plan slowly emerged: not to be content with rehashing and ordering the information already available, but to go to the source and look for new points of view. So I started a long round of interviews, and the more I talked to developers who were usually never considered by journalists and writers, the more I realised that there was a part of the story that had never been told.

Meaning?

Through the Moongate was the starting point for my reflection, which was followed by Video-Games, my new work. It is a five-volume work that traces the history of

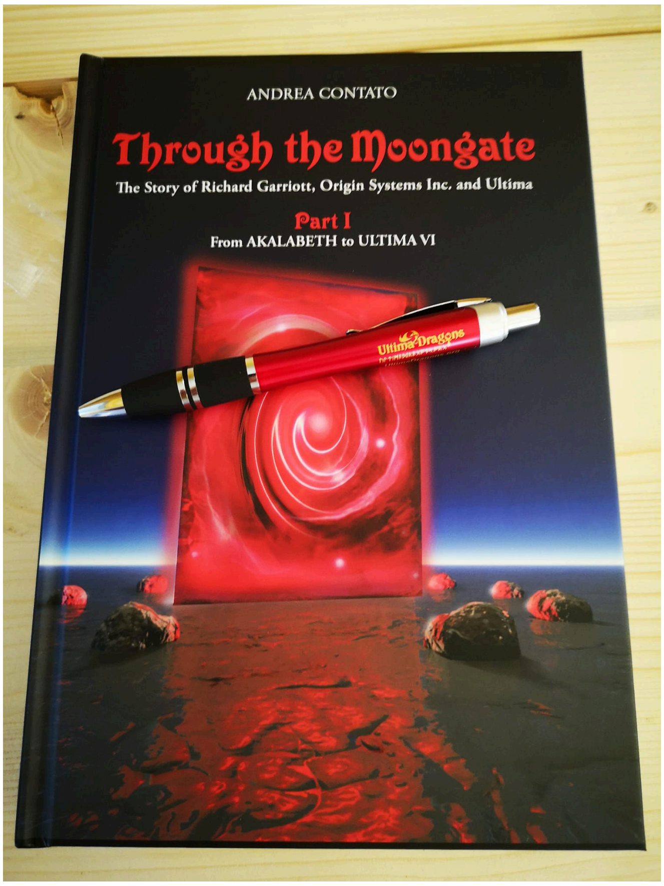

ANDREA CONTATO
Through the Moongate
The Story of Richard Garriott, Origin Systems Inc. and Ultima
Part I
From AKALABETH to ULTIMA VI
Ultima Dragons

Fig. 1 - Through the Moongate

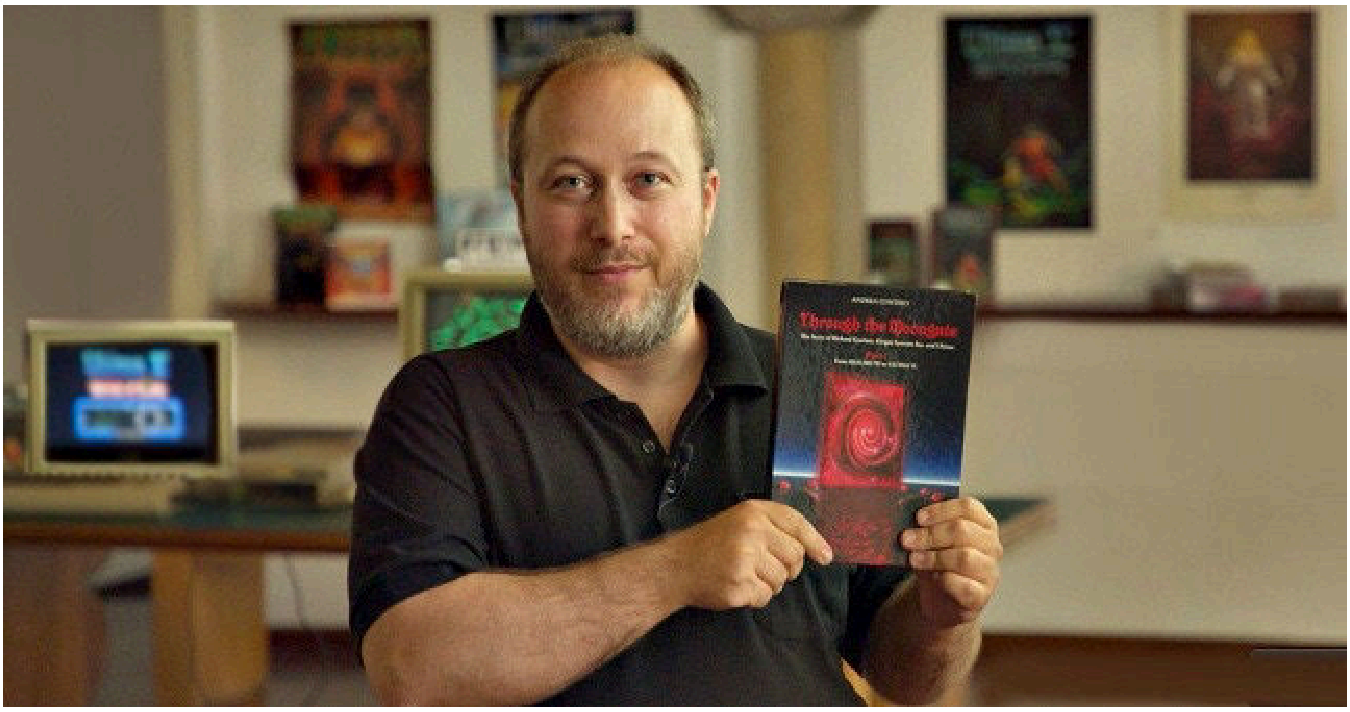


Fig. 2 - Andrea Contato with his book 'Through the Moongate'

video games from its origins to 1999, in its entirety (games for mainframes, minicomputers, microcomputers, consoles, arcades, handhelds), with a special emphasis on the human and creative aspect. In Video-Games I focus, as I started to do in Through the Moongate, on the developers with their experiences, their skills, their personal tastes, their relationship with technology and the choices they made in creating games.

Often when talking about the history of video games, the focus is on the economic issue, on competing companies, on consoles sold and cartridges sold. Video-Games is, first of all, a tale of human stories: people struggling with the creation of games; people who, often, before developing games, played those of others, had fun, became passionate and, in the end, decided to make their own game, often drawing inspiration from the work of others.

An ambitious work.

Yes, definitely. In fact, I gave up on compressing everything into one volume and decided to start with the idea of producing five. I hope it will be enough. Because it's fine to summarise, but there's a lot to tell and I'm not talking about anecdotes, but about information that, even if it's amusing and curious, is useful to paint a time that was and that won't be back, a way of making games that doesn't exist anymore and that tells us a lot about why what we consider today the great classics were made this way and not in another way, why they were successful and distinguished themselves from the rest of the market. It takes time and space to tell everything and, thanks to the fact that this is an independent product, I didn't have to come to terms with anyone.

Where can we find this book?

The first one is already available and covers the period from the origins of video games up to 1979. Subsequent volumes, which are in the process of being written and should come out annually, as early as 2022, cover five years at a time, from 1980 to 1999.

If you would like to buy a copy, you can order it directly

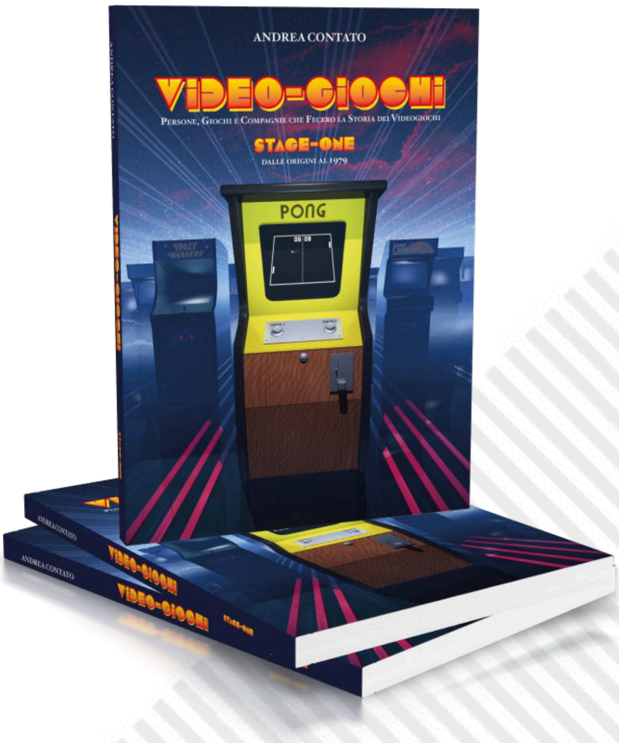

ANDREA CONTATO
VIDEO-GIOCHI
PERSONE, GIOCHI E COMPAGNIE CHE FECERO LA STORIA DEI VIDEOGIOCHI
STAGE-ONE
DALLE ORIGINI AL 1979
PONG

Fig. 3 - Video-Giochi stage-one

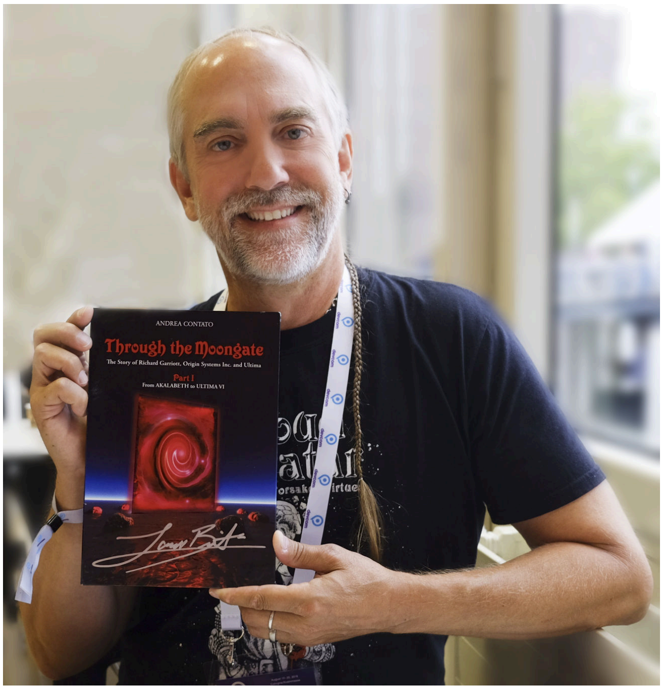


Fig. 4 - Richard with the book

from my official store at www.andreacontato.com/negozio, where you can also find the other books, or from the official Video-Games website www.retro-game.it or on Amazon. In the latter case, however, it is worth pointing out that the ebook version is only in MOBI format and that the Print on Demand paper edition is slightly smaller than the one printed by us.

Us?

Yes, Video-Giochi is published by Retroedicola Videoludica, a cultural association based in Bergamo whose aim is to disseminate and preserve the culture and history of video games.

Similar objectives to yours.

I would say so, in fact a fruitful collaboration was immediately born. Retroedicola has been scanning video game and computer magazines for years. Lately, thanks also to the remarkable talent of its members, it has embarked on the life of reconstructing some of the very first video games including Tennis For Two, OXO and Nimrod. There is an official website for these reconstructions, www.rebuildingbits.it.

How and when did your passion for video games begin?

How? By accident. When? In the early 1980s. My family and I were travelling around Europe in a Volkswagen Type 2 van, a real cliché. We stopped in Holland and I snuck into a bar on the campsite. There was an Asteroids booth that immediately caught my attention. I parked myself there for hours, maybe days, watching other people's games. When I tried to play it I immediately crashed and went back to being an enthusiastic spectator and ready to give suggestions, in Italian, to the international players. I must have been a real pain in the ass.

Then we went home and I didn't have any more experience with video games for some time. Arcade games were bar stuff, and bars in the early 80s were not places for pre-schoolers. But in 1983 or so a little Italian gem came into my house, a small home console. It was called Universal Game Computer and it was produced in Curno, in the province of Bergamo. Just a few kilometres from my house! But I didn't know that at the time. I discovered it many years later, which, by the way, allows me to anticipate one of my next undertakings.

How so?

Recovering the history of Cabel of Curno, one of the most important Italian companies operating in the video game sector. A small household appliance and television factory that also produced consoles for almost five years. And sold many that are now collectors' items.

What is the most difficult challenge for those researching the history of video games?

When I started a few years ago, I would have answered you: find the sources, find the people, manage to contact them, interview them, get the answers you need to fill in the gaps. In about ten years everything has changed. Some time ago, retrogaming and video game history were considered to be topics of little interest and many developers were wary when contacted. They didn't understand why there was so much interest in their lives and works.

Today everything has changed. There is a growing interest and the developers know it. When I contact them they are often very helpful, partly because many of them are now retired and elderly. So they have time to rummage through their memories, remember and tell their story.

So what is the answer today?

The problem is to be quick. The first pioneers of computer science and video games are already very old or unfortunately dead. It is necessary to hurry to preserve the memory before it fades forever. There is so much to investigate, so little time available. In our own small way, Retroedicola Videoludica and I are doing our best to preserve everything we can.

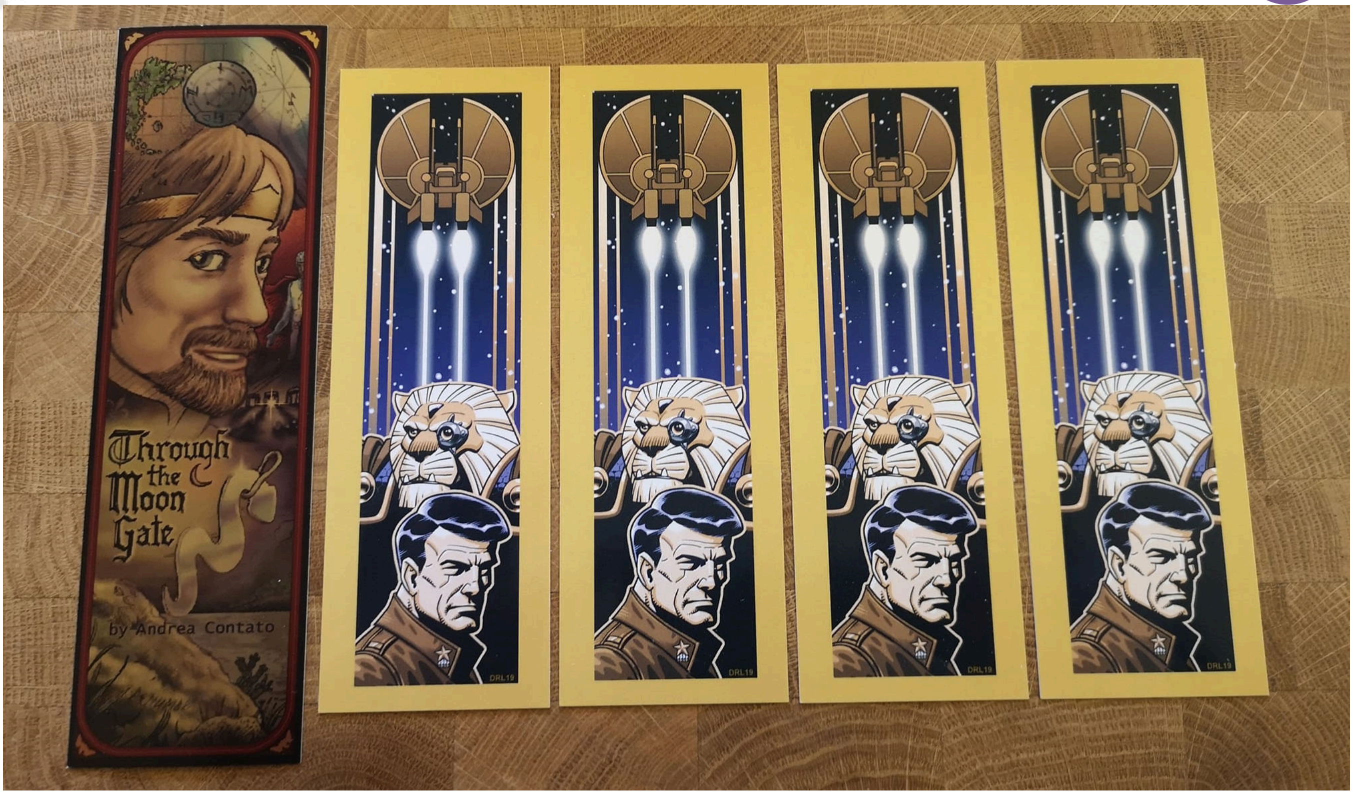


Fig. 5 - Bookmarks of Through the Moongate

Plans for the future?

Finishing Video Games. I have one volume to complete and three to write from scratch. The plan of the work already exists, although it is provisional and will surely be modified, and it plans to deal with all the most important games, companies and hardware up to 1999. We are also working on the English translation of the work. With Through the Moongate we have seen that abroad there is a lot of interest in these works and the fact that they are written by an Italian author is not a big obstacle. The language, however, is, and there is a lot of work to be done. By the summer, moreover, together with Enrico Ricciardi, who has already helped me to publish Through the Moongate, I am writing a book on the history of the early years of microcomputer games, from 1977 to about 1980, and to do so we have gone to find the pioneers who in those years laid the foundations for the computer games industry: Scott Adams, Mr and Mrs Williams, Richard Garriott, Robert Clardy and many others. In our opinion, this will be a special work because it was created with two different and well-matched goals: besides containing my historical research and interviews focused on the creation of the games and the technological problems they faced, it will have an incredible in-depth study edited by Enrico, in which the collecting and production aspect of these games - which have now become coveted and precious heirlooms - will be dealt with in great detail, with an unprecedented photographic section. We will do a Kickstarter, between March and May.

If the topic is of interest, we could talk about it again in the future.

Thank you for this interesting interview. I am sure we will hear a lot more about Andrea in the future, if you have any curiosities do not hesitate to contact him.

Useful links:

Official site: www.andreacontato.com
Official FB: https://www.facebook.com/Andrea-Contato-343821635963825
Twitter: https://twitter.com/xender76

Official site of Through the Moongate: www.theira.it
Tim's official FB: https://www.facebook.com/Through-the-Moongate-216842488907011

Video-Giochi Official site: www.retro-game.it
Video-Giochi official FB: https://www.facebook.com/RetroedicolaEditore
RetroEdicola's YouTube with lives on the history of videogames: https://www.youtube.com/playlist?list=PL2Iy4uB19ZctlnWL1YbEW-XvGRIBPYGrl

Kickstarter of LOAD AND PLAY: https://www.kickstarter.com/projects/1108065491/the-inside-story-of-the-birth-of-microcomputer-games

Let's take a look at Lykia - The Lost Island

by Carlo Nithaiah Del Mar Pirazzini

This project comes from the authors of Pets Rescue and Alpharay, two excellent titles.

Lykia is an action RPG that is very reminiscent of Super Nintendo classics such as Secret of Mana and The Legends of Zelda: A Link to the Past.

The game is coming to Commodore 64, Plus/4 and expanded C16 and will be available in digital and physical versions.

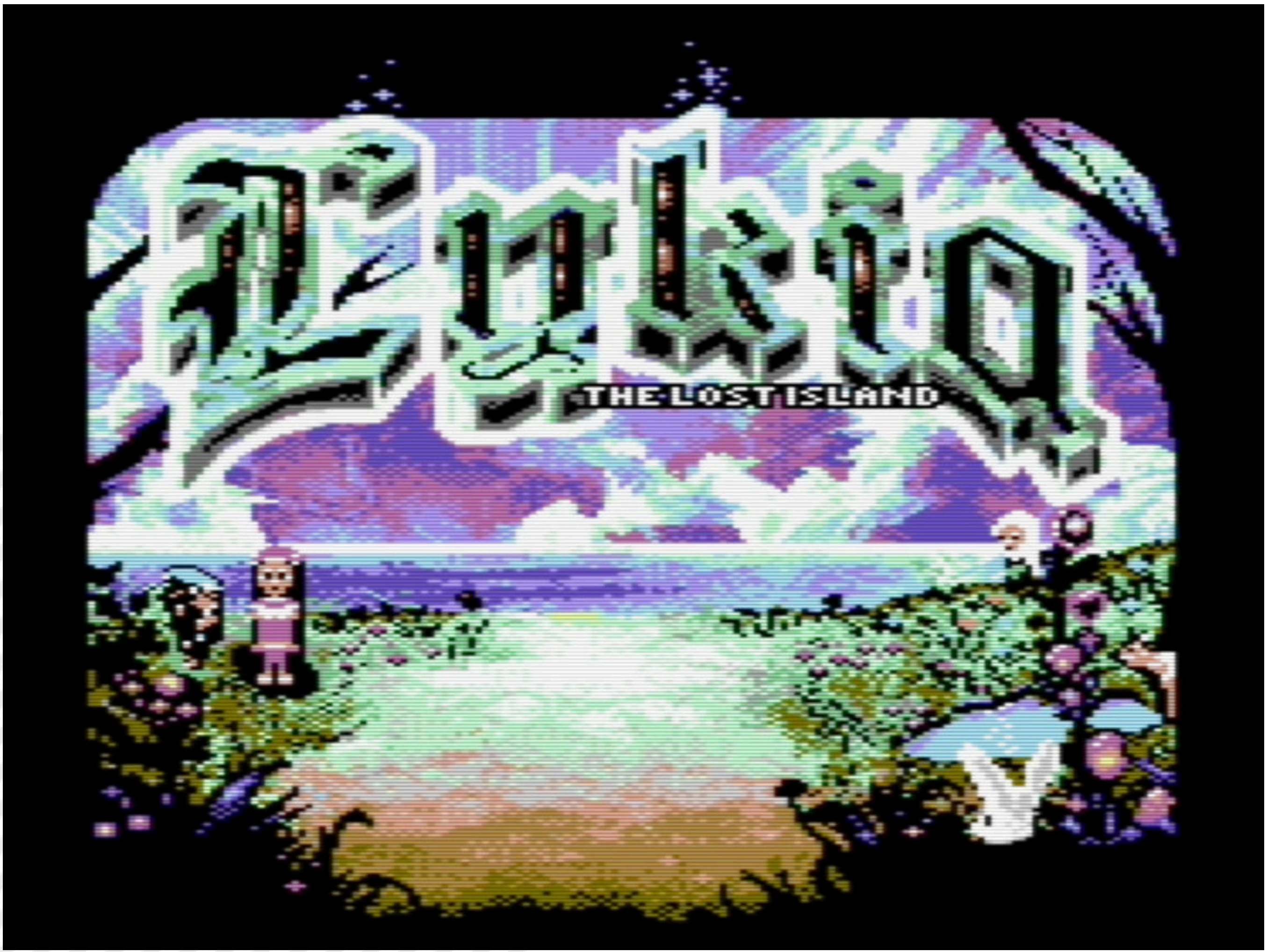
The story tells us about Nora and her 16th birthday, a party that will turn into something epic.

It all starts with an invitation from her mother to pick Apothia fruit, a very special plant. While Nora is picking the fruit, something happens that affects the whole village and the surrounding nature.

It also endowed Nora with magical powers.
In short, a classic introduction for this type of game.

We caught up with **Stefan Mader**, coder and mastermind behind the project, via social media and had a chat with him. Stefan is a very friendly and courteous person and sent us a full demo to have a look at the title, but more on that later. I leave the field open for Stefan to tell us about his title.

"The development started in about 2020... We thought with the team about a kind of game that you hardly ever see on Plus/4 and 8-bit machines... and I love Zelda! Things came together to create a test game to see if we could actually go ahead with the project and Lykia was born. In the team there is me from Germany, Robert Kisnemeth from Hungary who did the graphic tilesets, Ronny Doll for the music on Plus/4 also from Germany, as well as Markus Jentsch for the music on Commodore 64. Then there are



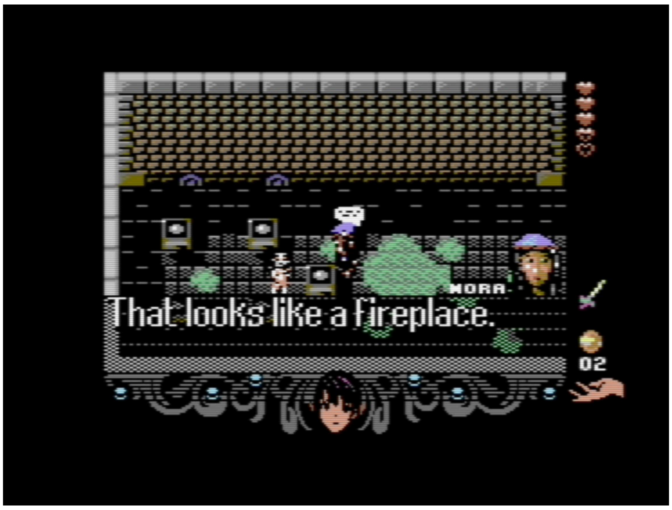
some developers who are in charge of running the framework routines like Ingo Jache, Imre Szell and Balazs Szabo. They take care of making the title as compatible as possible with flashcards, SD2IEC and others.

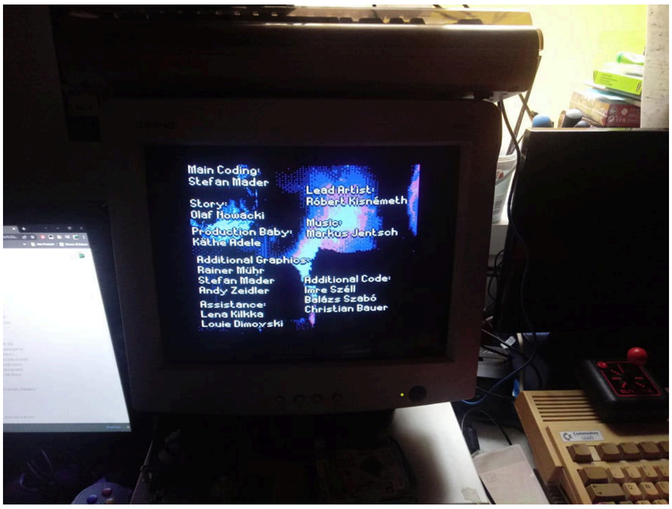
In fact, the team has been working together for quite some time. We are a well established team!

Speaking of release dates... we're actually almost done. I would say we are about 99% of the final product. However, the last 1% seems to take us a very long time. So unfortunately I can't give you a precise date for the release of the game. We do our best but it's a project in our spare time and most of us have families, work and other things to put in front... and I certainly can't force anyone to do things sooner.

But we are there. Have confidence!"

Stefan's words are clear and we confidently await the arrival of the finished version.





Talking about the actual game that we have seen, the work done so far is remarkable. A beautiful introduction tells the story of the game and is reminiscent of the Nintendo console titles.

The choice of graphics on both Commodore 64 and Plus/4 is also beautiful (the latter appeared even more 'vivid' to me in its colours and animations).

The soundtrack is very 'European' and you can hear it. It is a style perfectly integrated in the type of game but with a strong geographical connotation.

The menu system that can be accessed via the keyboard is also interesting. It's reminiscent of Zelda, and it's quite simple. You select the item and click the fire button to equip it. Simple.

The area to be explored is very large as are the enemies and dungeons. Beautiful.

We tested the game on emulation (VICE) and found no problems. As well as using it on a real Commodore 64 with SD2IEC and on THEC64.

So, we just have to wait for the final game to be published by Psytronik.

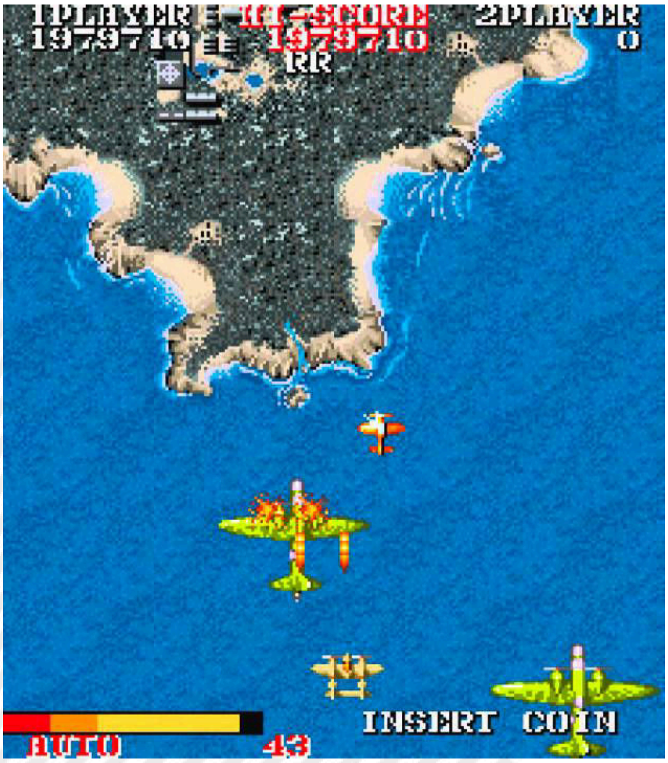
You can also follow the development here: http://www.psytronik.net/newsite/index.php/c64/140-lykia64

War and Videogames

by Mic the Biker Novarina

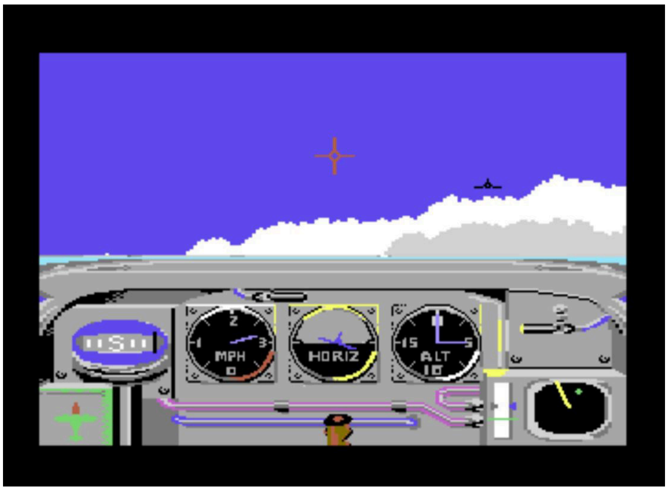
The beginning of the Second World War started, for much of the historiography, on September 1, 1939. I like to say it and always repeat it: "Historia magistra vitae est". It is our duty to remember these dates, to remember the evils that a war (whatever it is) brings with it. And we too, in our own small way, want to do it, but obviously in a different way. Everyone remembers the games set in the war period released in the last 15 years, both for PC and for the consoles that have followed. But war-related games have ancient roots that perhaps someone may have forgotten.

Many of them were inspired by events that took place during the war and the first two I want to remember are the immortals **1942** and **1943: The Battle of Midway** by Capcom. The Battle of Midway was a real flop for the Japanese fleet, which was saddled up for a massive operation in the Central Pacific aimed at the occupation of Midway Atoll. The Americans, however, were perfectly aware of the enemy move thanks to the deciphering of Japanese cryptographic codes carried out by the "Magic" system. The Battle of Midway between June 4 and 6, 1942 was the turning point of the war in the Pacific. Admiral Nagumo's four aircraft carriers were surprised by US bombers and sank within minutes. The first game of the series is 1942, released in arcade version in 1984 and then converted on a lot of platforms. It is a vertical shooter, as it was very fashionable at that time. The protagonist flies on a Lockheed P-38 Lightning with the aim of reaching and bombing Tokyo. During the long journey it will fly over various locations in the Pacific Ocean: Midway, Marshall, Attu, Rabaul, Leyte, Saipan, Iwo Jima and finally Okinawa. Obviously, our mission will not be without dangers, because of the great number of Japanese vehicles that shoot from all sides. In 1987 arrived the sequel, called 1943: The Battle of Midway. This one remains, in my opinion, the best of the whole series. He focused his plot on the Battle of Midway. The most significant novelty was the possibility of playing in two at the same time, which made this cab a real superstar of those years. It was really a crazy thing to stand shoulder to shoulder with the best friend and together set up blasting tactics while performing the famous loop!

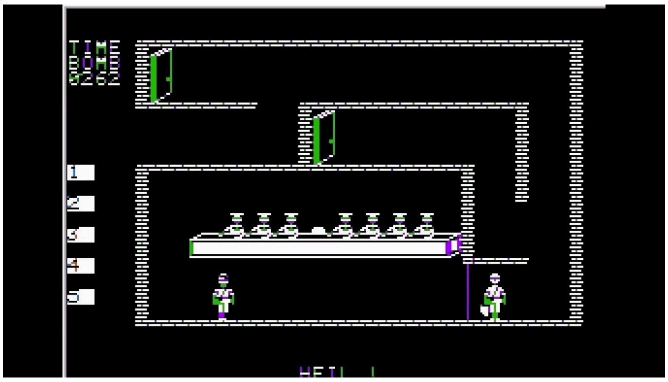


1943: The Battle of Midway

Meanwhile, even on home computer the topic was hotter than ever. A great game was **Aces of Aces**, released in 1986 for all 8-bit platforms by three different companies: Accolade, US Gold and Tiertex Design Studios. Ace of Aces is a combat flight simulator set in World War II. The player flies long-range on a fully equipped fighter-bomber. On board we have missiles, bombs, and a machine gun. The missions include the German fighter planes, bombers, V-1 flying bombs, U-boats and trains destruction. There are five different display options: the cockpit, the right



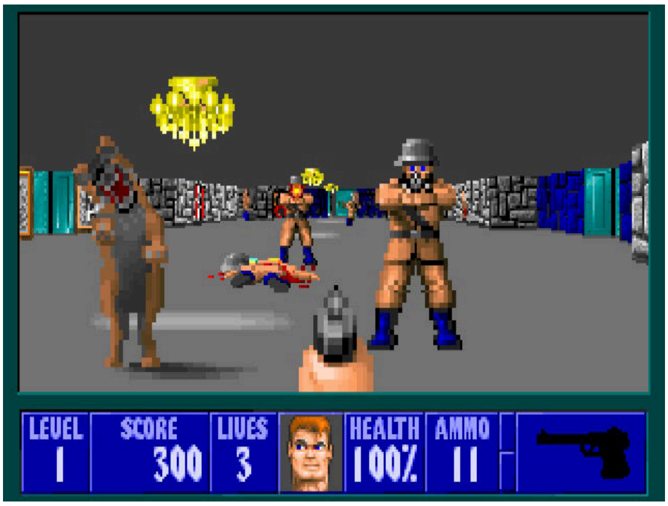
Aces of Aces



Beyond Castle Wolfenstein

and left wings, the navigation map, and the bomb bay. All of them are selectable using the keyboard or by double-tapping the fire button and moving the joystick in the desired direction. It was one of the best-selling games for Commodore 64, the second ever by Accolade, and still today it's a pleasure to play. This is because the simulation component was present but is not heavy: during the missions, we started in flight, eliminating the stress of raising a bomber from the ground and, above all, make it land without damage.

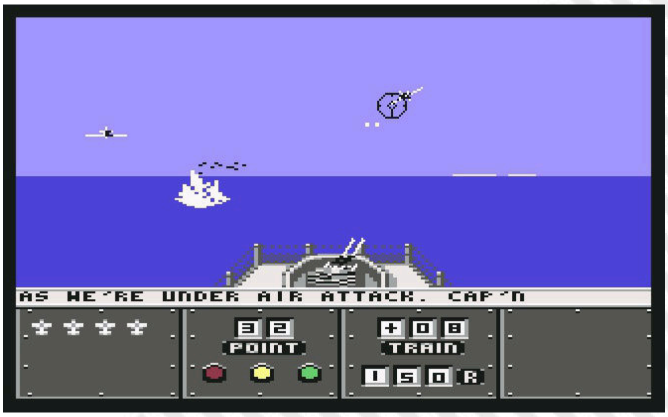
I'm hearing many of you grumble: I am sure that everyone, talking about world war II, have a think at it, **Wolfenstein**! Nothing to complain, you are damn right. The first incarnation of this immortal saga came from 1981: perhaps not everyone knows that this masterpiece was born on Apple II. The game, which takes the name of Castle Wolfenstein, has a very simple graphics, and adopts a view from above but with representation of the characters in profile. What we will have to do is wander in the castle, consisting of 64 rooms, in search of secret documents. In each room we can find chests containing useful objects for the continuation of the game, but also enemies ready to take us out. These can be either simple soldiers, easy to shoot down, or the deadly SS, equipped with bulletproof vests, more difficult to defeat.



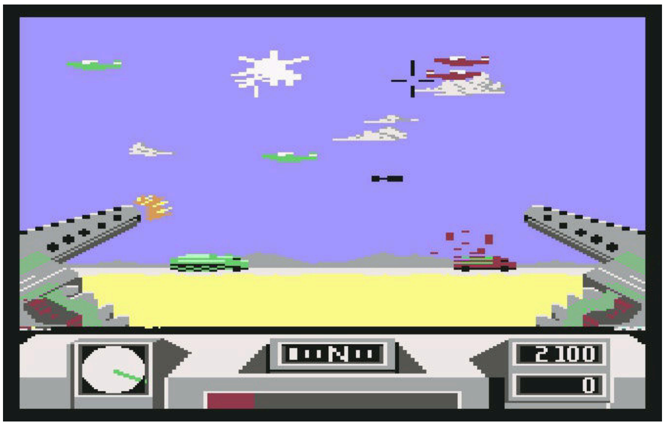
Wolfenstein 3D

The official follow-up came in 1984 under the title **Beyond Castle Wolfenstein**. It was initially produced on Apple and later ported to Atari 8-bit and Commodore 64. Here the plot focus on a very specific historical fact: the attack on Hitler in July 1944. In the game we will have to cross the Furher's mazy secret bunker, to reach the room where he holds his meetings, in order to place a bomb outside it. But this title hits the point in 1992, when it became a first-person shooter for PC! Called **Wolfenstein 3D**, it takes us in charge of the legendary William "B.J." Joseph Blazkowicz, an American soldier who is prisoner in the famous castle. The game consists of several levels, obviously infested with armed guards and ferocious dogs. The castle is more intricate than the labyrinth of Minos, turning it without a map is really hard. In each level there are secret rooms full of treasures, food, first aid kits and different types of weapons. What made this chapter immortal was its innovative graphics engine and first-person view. The pseudo-3D technique had already been used in a couple of titles but here the level rises a lot, bringing much more detail and colors to the screen. It was something decidedly pioneering, scrolling was fast but unnatural and often caused nausea or discomfort in the eyes. In the initial plans of the programmers the stealth aspect should have been more present, but these more simulative modes were abandoned since they would have made the game less dynamic and too complicated to manage.

The naval battles were also taken over and analyzed to create remarkable games. **Destroyer**, for example, is a great simulator, published in 1986 by Epyx for commodore 64, Apple II and the various IBM compatible. In the game



Destroyer

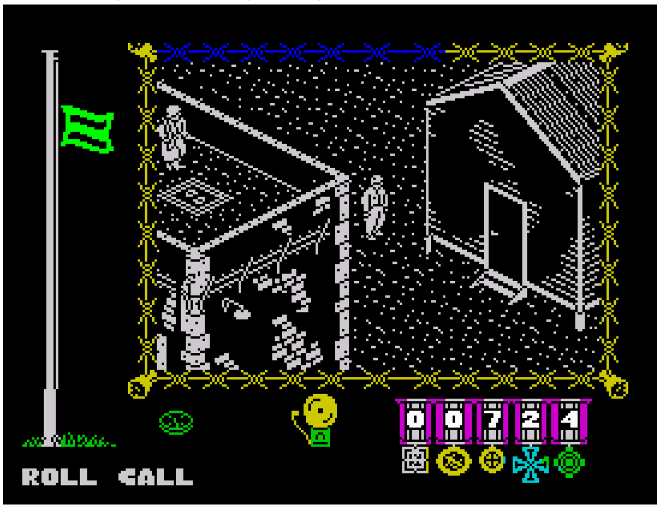


Desert Fox

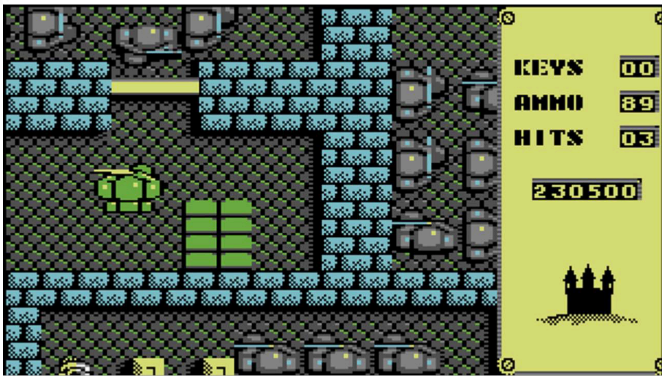
we will find ourselves in command of a Fletcher Class destroyer, in a typical context from the Second World War, while not referring to any event. It is not a simple game, since we will have to manage both the navigation than the combat, complete with manual aiming of the weapons. To make you understand the difficulty, we will have to move between 13 different screens, each representing a station. To change location, we will have to enter a two-letter code, which will start loading a part of the game. The action takes place in real time, it does not matter which screen we are on: every important event will be reported on a message bar. A true masterpiece not only for those years: also today playing with it is a challenge. The Amiga version was not so good: released in 1988, is a transposition of the one for Commodore 64.

And what about the sublime **Desert Fox**? It is a strategy and action video game, released in 1985 for the Commodore 64 and in 1986 for the Amstrad CPC. This time we move to the desert, more precisely to North Africa. We are aboard a powerful Allies M4 Sherman called "Lone Wolf", and we will have to try to fight many missions against the forces of the German general Erwin Rommel, called "desert fox". In the campaign mode the context is that of 1941 in Libya: we are in the strip of land between Tobruch and Sollum and our task will be to protect the allied deposits from Rommel's attacks. On our map we will see only the deposits marked by a flag, our chariot, and a swastika to indicate the German general. Enemy troops are not reported, we will have to use our brain and strategy to intercept their communications, in order to understand their position and avoid them. When we have the perception of being in serious trouble, we can call an air strike to gain some time. When we reach a deposit, it will be considered safe and we will receive a full tank of energy. It will be easy to come across some enemy vehicle, or Rommel himself since his chariot is constantly on our tracks. In these cases, you will switch to combat mode, with a first-person view. Nothing is left to chance, even the sequence day and night is respected, and I assure you that facing a battle in the dark really makes anxiety. The Game Over will come if a deposit is conquered by the Germans or if our tank is destroyed.

Staying on the 8-bit world, in 1986 came out **The Great Escape**. The year is 1942 and we will have to lead a war prisoner to escape from a German concentration camp. The game environment is the prison camp with isometric view and multidirectional scrolling. There is only one entrance to the camp, ultra-guarded: the passage is allowed only if in possession of the correct documents. The other sides of the prison camp give on steep cliffs that thrown into the sea. The action starts from the dormitory, shown on a fixed screen, from which the day starts: it has a daily routine consisting of the appeal, exercise, meals, and bedtime. This rhythm is the same for all prisoners, who will follow this process; if we leave our character still, after some time he will adapt to the routine. Obviously, the camp is surrounded by fences and guard dogs, used to patrol the perimeter. Moreover, we add the guards in the towers, to observe any prisoners trying to escape. In all this context we must try not to get caught around the field in search of the objects useful to escape. One flag represents the level of the prisoner's morale: if it goes to zero, the game will be over. The color of the same warns us if we are in a dangerous area (red flag) or quiet (green flag). We can only carry two objects at a time, so you must think carefully about what to bring with you each time, regarding the actions you want to take. For example, entering the underground tunnels without light is a very dangerous task. We will therefore



The Great Escape



```
KEYS    00
AMMO    89
HITS    03
230500
```

Into the Eagle's Nest

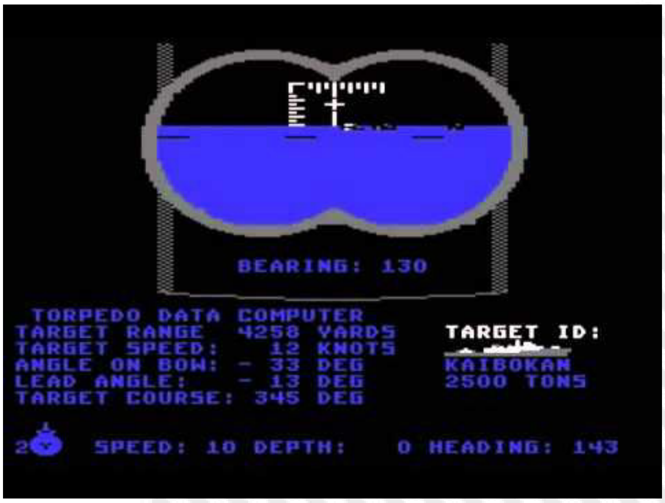
have to hide around all the objects found, which may be useful to us later. If this is not enough, the guards could search us and arrest us in the red flag areas, while the captain of the camp can do this always and anyway. I'm honest, I've played a lot, but I've never been able to escape from the field once.

One game that literally drove me crazy was the colossal **Into the Eagle's Nest**. Released in 1987 for the 8-bit platforms and shortly after also for the first 16 bits, the game takes up the dynamics of Gauntlet and transports it to the Second World War. The title is freely inspired by the building built on the top of the Kehlstein, which connects with the villa below called "Berghof". The building in question was never used for military purposes, but only for meetings and diplomatic stuff. But its posthumous fame and high-sounding name, coined by the British journalist George Ward Price and clear reference to the ancient symbol of Germany, were an opportunity too greedy not to make a game of it. The action takes place inside the fortress, divided into several floors, with a view from above and multidirectional scrolling. The environments are full of German soldiers who can damage us by touching, while we have a machine gun with a maximum of 99 shots to defend us. We must free three allied prisoners, which are the first three missions of the game: once saved, the soldier will follow us until the end of the level. The fourth mission is the most difficult since you will have to activate the explosive charges necessary to destroy the fortress and escape in time. For me it was an absolute masterpiece!

Among all this great games cannot miss one concerning submarines. In this box I would like to put a diamond of rare beauty, **Silent Service** of 1985. We are talking about a submarine simulator produced by Microprose for the major 8-bit home computers, programmed by Sid Meier, while in 1986 it was also released on 16-bit systems. A game so realistic as to create some problems in West Germany, here titled Das U-Boot. The setting is the Pacific Ocean and the period is World War II. The name of the game comes from the nickname of the US military submarines stationed in the bases of the Pacific Ocean. The game is inspired by the battles that took place in the middle of the ocean between American submarines and Japanese ships: the aim is to sink the Japanese merchant fleets so that they cannot supply the troops of the rising sun. Here, as in Ace of Aces, the dynamics unfold through multiple display screens: maps, dashboard, interiors, periscope when submerged and turret when emerged. The accuracy is incredible, with the weather changing sometimes affecting visibility, and the day and night cycle, which allows us to use the dark for surprise attacks. Silent service is crazy, we can use the tactics used in the conflict, including the possibility of going deep into the seabed, to avoid the Japanese depth bombs. The attention to detail is obsessive: the maps are realistic in every detail, including the depths of the sea. The weapons that can be used faithfully follow the real ones, as well as their characteristics: for example, torpedoes leave their wake in the water and can be defective, with the problems that derive from them. Japanese destroyers can detect the presence of the submarine using sonar or simply by observing the surface of the ocean: if the submarine is traveling at shallow depths, it will leave a trail easy to spot. In a nutshell, a masterpiece that deserves to be rediscovered.

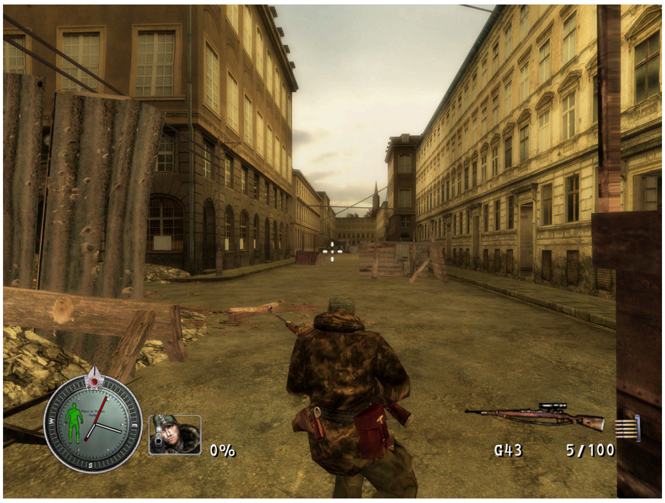
Modern times

And then we come to the present day, where PCs and Consoles are increasingly powerful and allow you to make almost cinematic video games. The Call of Duty and Medal



```
                BEARING: 130

 TORPEDO DATA COMPUTER
TARGET RANGE    4258 YARDS      TARGET ID:
TARGET SPEED:     12 KNOTS      KAIBOKAN
ANGLE ON BOW:   - 33 DEG        2500 TONS
LEAD ANGLE:     - 13 DEG
TARGET COURSE:  345 DEG

2    SPEED: 10 DEPTH:     0 HEADING: 143
```

Silent Service



Sniper Elite

of Honor sagas are compelling and increasingly detailed. The historical facts are re-proposed with an accuracy that will leave you breathless. Those who are thinking about the Normandy landing of **Medal of Honor** have right understand: incredible atmospheres, you find yourself throwed into the midst of the frenetic and crazy action with the only thought of bringing home alive, while from the bunkers on the beach it rains tons of bullets. But for me among the "new" games on the theme of the Second World War there is one that I consider unattainable, unique, magnificent. Almost perfect.

I'm talking about **Sniper Élite**, released in 2005. We play the role of Karl Fairburne, a US agent of the Office of Strategic Services in German guise. The character fits into the Battle of Berlin in 1945, during the last days of the conflict: On April 16, 1945 the Red Army launched its last general offensive, with the objective of Berlin. The losses were very high and the decisive breakthrough, obtained by the brute force of thousands of tanks, was obtained only on April 20. In this context fits the plot of the game, which sees us engaged in disguise to obtain data on the German military nuclear program before the Soviet Union does. Throughout the course of the game, we will have to deal with different factions that compete for control of the city: the German resistance, which comes to our aid, the Soviet NKVD, and the Nazi forces. During the continue of the plot, we will find ourselves in front of historically existed characters, such as Martin Bormann, an extremely important figure of the Reich. He will be one of the goals that Karl will have to eliminate! Unlike the totality of the other most recent games on the genre, in Sniper Élite moving in stealth mode is the only way to proceed in the game. Observing the scene well before taking a step makes it clear how terrifying the situation was. You spend most of your time crawling or moving on your knees in search of shelter. The realism of the aiming system is crazy: taking your breath, observing rags, flags or leaves on trees to understand the direction and intensity of the wind is the basis for a precision shot. The point at which the projectile hits the enemy has several effects, from the light wounding that will alert the soldier to the heavy wounding. For example, an enemy wounded in the leg will crawl to seek shelter or sound the alarm. A blow to the head is simply one shot, one kill. Sniper Elite was awarded in 2005 by The Independent Games Developers Association as "Best PC/Console Game", and it could not be otherwise.

The War is over

How World War II ended we all know: On August 6, the B-29 Enola Gay bomber dropped a uranium bomb (Little Boy) on the Japanese city of Hiroshima. Three-quarters of the city was destroyed, and 78,000 people died instantly. Three days later, on August 9, the B-29 Bockscar dropped a plutonium bomb (Fat Man) on the city of Nagasaki. Two-fifths of the town was wiped out and the immediate victims were 35,000. But many thousands of people perished in the following days due to severe burns and radiation poisoning.

It is indeed very important to transmit and keep alive the memory of these terrible historical facts so that they never happen again.

And if moreover I managed to make you rediscover some forgotten titles... Well, then for me too it is mission accomplished. Over and out.

AMIGA: the top 10 (+1) game intros

by Beppe Rinella

Welcome back to all you friends who, like me, were very disappointed when you discovered that the Commodores were playing ordinary musical instruments instead of wonderful Amigas!

As we all know, there was no time to waste in the arcade, and once you inserted the token you were right in the thick of the action.

There were not many titles that included a small introduction telling what was going on, but there were.

Just like that, Ghost'n Goblins comes to mind, where we helplessly witness the kidnapping of our beauty whom we have to free, or Double Dragon, where basically the same thing happens.

It was only a few seconds and often with 'in game' graphics, nothing even remotely cinematic.

Some titles, later on, did a bit more, like Final Fight for example, where it was possible to see the intro in the rare moments when the cab was free.

The reasons for this were obvious, the arcade was a place to play, there was no time to waste, the queue was long and there was plenty of money to spend.

With the home game, however, the matter changed and a lot, we had plenty of time to enjoy the story we were going to take part in, who the protagonist was, what was happening and why.

The narrative component became more and more important, and with time and increasingly powerful consoles and personal computers, the intro became more and more impactful, in some cases highly spectacular.

Another World

All this to present this month's list:

THE 10 (+1 bonus) MOST SPECTACULAR AMIGA INTROS

Mind you, by 'spectacular' I don't just mean the aesthetic aspect, which as we shall see in some cases may not boast grandiose effects, I mean the whole thing... but in short, we understand each other.

ANOTHER WORLD

I talked about it at length a few issues ago, about the game and, inevitably, its intro. The opening sequence of Another World is simply perfect from any point of view, if there was a manual on how a story should be introduced, Another World would have a dedicated chapter.

Starting from the aesthetic aspect, graphically exceptional, with a wonderful 3D and spectacular animations thanks to the use of rotoscope. The sound is not less, a few effects but very appropriate (I'm sure that many of you, if not all, are able to "hum" the sequence of keys pressed by the protagonist to access the laboratory). But above all it is the music that contributes massively to the final result, creating the right suspense and making us perceive right away that something big is going to happen and it is not good.

What, in my opinion, makes the introduction of Another World magnificent is the fact that we live together with the protagonist what is happening, unaware as he is, we are told absolutely nothing.

A total lack of information only adds to the sense of disorientation that accompanies us throughout our adventure.

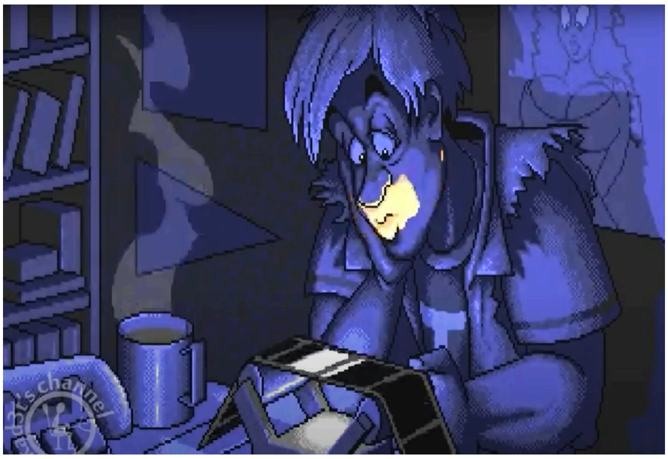
Never before had we seen such a close encounter between video game and cinema, simply a piece of history.

PREMIERE

A truly delightful game Premiere, the presentation that introduces the protagonist and what's going to happen to him is just as delightful.

Clutch, this is the name of the character we will be asked to guide, is a film editor

cinema. During the night shift he falls asleep and the films



Premiere

he is working on, which have to be ready for the following evening, are stolen. In the morning, after being woken up by a phone call from the boss, he discovers the theft and sets off in search of the lost films.

All this is told to us through a delightful little animated short, with a very cinematographic style and therefore perfectly in line with the main theme of the game. The animated sequence is nothing more than the beginning of the film "Premiere", complete with opening credits, including the name of the protagonist, who is then introduced as an actor.

Everything is divinely realised, an animated cartoon with all the trappings, designed and set to music to perfection. After all, Premiere was developed by Core Design, who knew their stuff when it came to animated and cartoon-style intros, and had already amply demonstrated this with titles such as Chuck Rock, Heimdall or... we'll get to that in a bit.

Whenever I decided to play Premiere, I don't think I ever skipped the intro, which was so beautiful and got me in the right mood for the adventure with Clutch.

CANNON FODDER

One of the most iconic intros to one of the best titles in Amiga history.

When it came to Sensible Software, everyone knew not to expect spectacular graphics and effects. Cannon Fodder and its presentation was no exception, relying entirely on irony (too much for some, but I'll get there) and a song that once it gets into your head will never get out again. In this case the introduction is not intended to tell the story and characters of the game, but to introduce the team that developed it. This is done by means of a number of photos, accompanied by the name and character 'played', of the individual members of the group, all dressed in military clothing. Like "STOO as Elvis" for example, with photos of Elvis Presley also in military clothing.

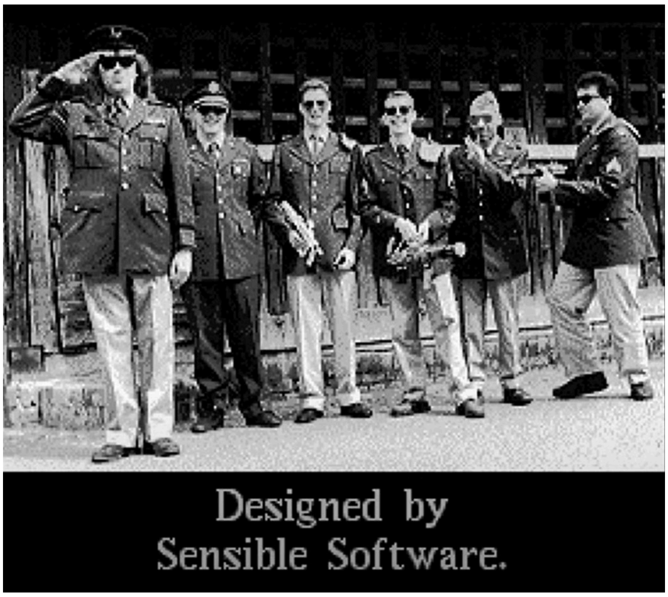
Visually entertaining but not at all spectacular.

But it is with the accompanying song that the sensible boys achieve their masterpiece. 'War Never Been So Much Fun' is a great little gem, with its upbeat rhythm, horns and warm female voice singing about how war has never been so much fun.

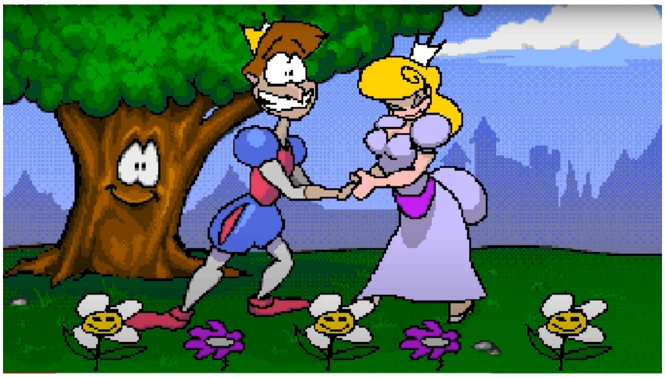
Composed by Jon Hare and the late Richard Joseph, this was not the only song they wrote for Cannon Fodder, they also wrote 'Narcissus', the theme we can hear when we see the hill with the graves of the fallen, poignant and wonderful, with Ian Ritchie, who played with a certain Roger Waters, among others, on saxophone.

Coming to the irony that many considered excessive at the time, Cannon Fodder had quite a few problems in Albion land. All because of the poppy that we see in the intro (and that initially appeared on the cover of the game), which is not just any flower, it is in fact used as a symbol of the Royal British Legion, a charity that supports war veterans. If we then add the title of the

main theme and the general tone of the game, where you have fun killing soldiers here and there, the controversy is served. It was more than a controversy, in fact, it went to court but everything was resolved with Sensible paying £500 for the rights to use the poppy. Definitely a nice gain for the English development team, given the large amount of free publicity that the whole thing gave to the



Cannon Fodder



Superfrog

game, which in fact (but not only for this reason, we know) sold very well.

This explains the disclaimer in the intro.

On a final note, the Jon Hare/Richard Joseph duo evidently took a liking to it and together composed the themes for Cannon Fodder 2, Sensible Golf and Sensible World of Soccer.

SUPERFROG

Speaking of the best intros ever seen on the Amiga, it's really impossible not to mention Superfrog.

The story it tells is not particularly original, nor does it want to be: the classic prince and his equally classic princess annoy the (also) classic wicked witch, who just won't tolerate so many niceties. So she decides to kidnap the princess, but not before transforming the princess into a witch.

prince into a frog, leaving him so sad and disconsolate on the riverbank. Just the river will bring to our (almost) hero a bottle of Lucozade (compliments to the dirty people who threw it away eh!), drink unknown to us as well as sponsor of the game and that he will drink without thinking too much, thus transforming himself into Superfrog, with the unfailing red cape of a true superhero. All this is told through a sublime animated short, masterfully realised from every point of view. Colourful, beautifully animated, cheerful and with a couple of gags that bring a big smile, like the technician who forgets to turn on the fan, thus not making our hero's cloak flutter in plastic pose (thus revealing how everything we see is a simple film set). Or the foolishness of our frog when he tries to take flight, which is so Ralph Supermaxieroe.

The author of all this was the American cartoonist Eric W. Schwartz. Schwartz, with his style definitely inspired by the Looney Tunes, who among other things created (using an Amiga) the character "Amy the Squirrel", a female anthropomorphic squirrel that reached such a level of popularity that it was "adopted" as an unofficial Amiga mascot.

Superfrog, rightly or wrongly, was (and is) often compared to Sonic and Mario, true masterpieces as well as yardsticks for any platformer of the time.

Neither of these two titles, however, could boast an animated presentation of such beauty.

FLASHBACK

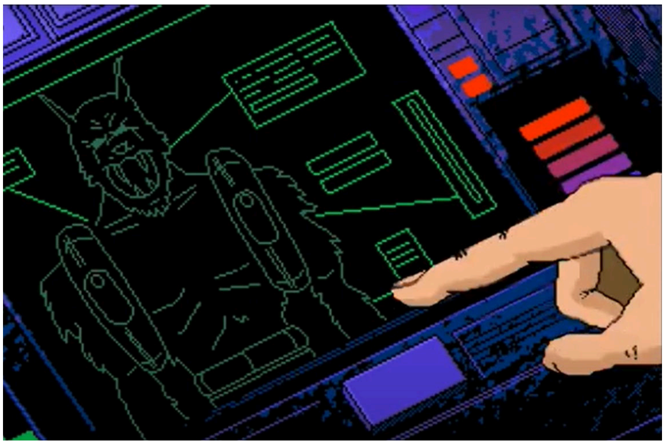
A magnificent title, to say the least, which owes a great deal to its predecessor Another World, which came out only a year earlier, a "mere" spiritual predecessor, though, considering that the two masterpieces are not connected in the least.

The legacy left by Another World, picked up in full by Flashback, is already clearly visible in the sequence that introduces the story.

Starting from the graphic realization, we find also here a massive use of animations realized with the "rotoscope" technique, used also for the in-game animations. After the intro we see a chase, a man is running away (which must be the protagonist because yes, the protagonist is always the one who escapes) chased by ugly mugs armed with a laser gun, which they do not hesitate to use against him. The fugitive climbs aboard a futuristic flying motorbike and continues his escape, but the bad guys don't waste any time, they climb aboard their spaceship which is decidedly more efficient, they reach the unfortunate man in a very short time and they shoot him down without too many compliments with laser shots, making him fall in the middle of a forest.

As in the introduction to Another World, we know nothing about what is happening in front of our eyes, and after the animated sequence we have more doubts than certainties.

Flashback



Wolfchild

Who is that man? Why is he running away and from whom? Where are they? The answers will only come by impersonating the mysterious man and completing this fantastic adventure. An exciting introduction, full of action and pathos, thanks also to the excellent music that accompanies it all, truly a little gem.

The adjective "small" is not accidental: the duration of the animated sequence is exactly 40 seconds, more than enough to make us want to grab the joystick and find out what the hell is going on.

WOLFCHILD

Once again, Core Design proves that it knows how to get things done when it comes to animated intros.

Kal Morrov, a world-famous scientist, is kidnapped by the criminal organisation Chimera, which wants to exploit the doctor's incredible discoveries, which allow the creation of hybrids between humans and animals. Needless to say, the aim of all this is to form an army of soldiers with extraordinary strength and skills.

Doctor Saul's son, the protagonist of this great title, decides to face the fearsome Chimera and free his father, but he can't do it as a simple man (even if he is very physical), so he decides to activate the Wolfchild project to face such a mission.

The intro shows the troubled and decidedly angry protagonist, alone in his mountaintop home, experimenting with his father's discoveries, turning into a fearsome werewolf.

The presentation is of the highest level, graphically excellent, the desire for revenge and revenge of our hero is palpable, after the exciting transformation we are waiting for nothing more than to impersonate our hero and punch anyone who stands in our way. They have no idea who they are up against.

A great "origin story" worthy of the most classic superhero. A frighteningly angry and revenge- hungry man who transforms himself into a very strong wolf-man, tell me what could be so exciting!

EPIC

If we want to sum up the story of Epic in one sentence, we can say that it tells the biggest mass migration in (video game) history.

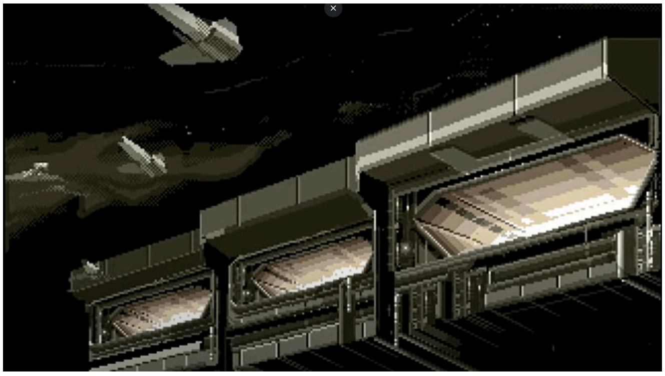
The star that lights and warms our planet is destined to die soon, turning into a devastating Supernova. Our civilisation's only hope of survival is to colonise the only planet judged fit to house us, Ulysses 7.

The problem is that to get there we will have to cross the territory of the Rexxon Empire, an alien civilisation always ready to start a war and with which every diplomatic attempt to avert a bloody battle has failed miserably.

The only hope of survival against the superiority of the evil empire are three spaceships made of Epical, a particularly resistant material. One of these will be the one we will be asked to pilot (the other two will escort the migrant population) to fight the evil enemy.

In the animated intro we are told all this by showing us the great exodus, a dramatic moment underlined by the excellent music, epic and threatening as the journey our people are facing. Graphically it is really a joy for the eyes, a combination of static landscapes, very well made, thanks to which we can admire our planet of origin as well as those we meet along our journey. Landscapes that are the background to the imposing fleets in movement, realized with an excellent 3d for the epoch (we are in the 92) really remarkable.

The Epic sequence is everything every fan of Star Trek, Star Wars, distant galaxies and anything else involving



Epic

Aquaventura

space battles wanted then and now, and I am definitely among them.

AQUAVENTURA

The presentation of Aquaventura is probably the best example of a red herring in the history of Amiga games. Its incredible quality was as great as the disappointment with the actual game.

But we're talking about intros without worrying about what happens next, and Aquaventura's intro is perhaps among the most spectacular ever seen on our beloved home computer.

The story of Aquaventura, which is not told through the animated sequence, is very simple: our beloved planet is devastated by a nuclear war, the few survivors have formed eight underwater colonies, until a hostile alien race decides to invade the planet, occupying all the colonies. The few survivors live in a spaceship orbiting in space, from where we set off in our ship to return to Earth in an attempt to recapture the Earth colonies.

It is from this moment that the intro starts, we see the departure of our little spaceship with Josh Aldrin, our alter ego, on board.

As already mentioned, the animated sequence is technically impressive, created with ray tracing that was absolutely stunning for the time. Remember that Aquaventura was released in 1992, but the intro began to circulate well before that, the game was announced four years before its release, one year after its release.

enormous period of time by the standards of the time.

During the presentation we can admire our little spaceship, with a design that still makes a great impression today. Inside, as mentioned, we see the pilot nodding his head to give the OK, the engines explode with all their power, and off we go at great speed into deep space, not before literally evaporating a guy who was far too close to the runway.

The sequence is anxious, dark and oppressive. Everything is very dark and there is no music in the background, just a few beeps, the roar of the muffled engines (we hear it from inside the ship) and the final explosion. Yes, because once started, we will see behind us the "mother" spaceship move away and explode shortly after, making us the last surviving human being.

In short, Aquaventura boasts a top-notch presentation, graphically superb, which perfectly creates the atmosphere of anguish one can feel just before leaving on a desperate mission. It is a pity that the game did not live up to this.

SHADOW OF THE BEAST 2

We were still missing the vision of something definitely leaning towards horror, weren't we?

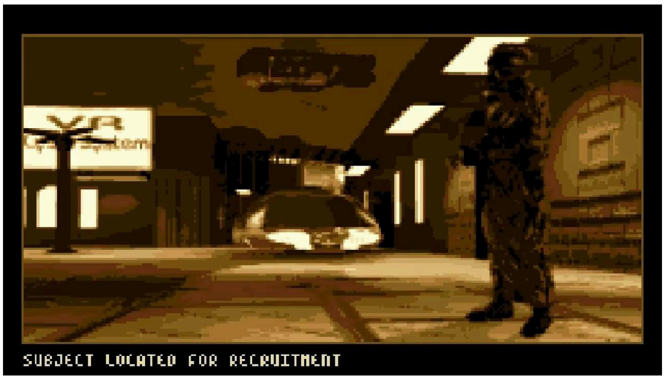
We recover endured with one of the intro that has made history on Amiga, that one of the second chapter of Shadow of the beast. A title in my opinion almost unbearable, of a totally senseless difficulty, but it is undoubtedly that the saga is among the most famous ever released for our beloved Commodore.

Zelek, the necromancer servant of Maletoth, the villain defeated in the first chapter, kidnaps the protagonist's sister in order to use her in who knows what ritual, thus restoring her master to his physical form.

What we see is the moment of the abduction of the poor child, we see the house lost in the void from the outside, Zelek observing it from afar and who thanks to his magic is transformed into a winged monster. From outside the house we hear the cry of a little girl, Zelek swoops down on the roof and, from inside the house we see his big hand which, after having destroyed the roof, kidnaps the child, amidst the heartrending screams of her mother.

Graphically the animated sequence is truly remarkable,

Shadow of the Beast 2



Syndicate

the dark and horrific atmosphere is rendered to perfection, the external setting not at all reassuring and desolating, the house bare and lonely.

The soundtrack makes a great contribution to this, the thunder rumbling, the child's crying, the woman's desperate screams. It all works perfectly.

The intro to Shadow of the Beast 2 is a small masterpiece, especially considering the year of its release.

Unfortunately, the game was not humanly bearable, terrifying even, but in this case that is hardly a compliment.

SYNDICATE

Anyone who has loved and still loves Blade Runner (present!), cyberpunk aesthetics and dystopian futures, could not help but deeply adore Syndicate, an authentic masterpiece created by the brilliant minds of Bullfrog.

In a hopefully very distant future, the world is divided into three large areas, each controlled by a large corporation. The aim of each corporation is to prevail over the others and thus manage to rule the entire planet.

Each of them has its own army of mentally controlled cyborgs, which are made to perform the missions necessary to achieve the final goal.

Once we have chosen a guild to join, we set out to conquer the world without any qualms in this masterful real-time strategy game.

The intro that introduces Syndicate tells us absolutely nothing about this, it was all told in the manual/book of the game, which obviously yours truly did not own.

What the presentation does do is give us an immediate feel for the game's atmosphere, as oppressive and dark as any self-respecting dystopian future.

We see floating machines, a huge flying vehicle and an unsuspecting bystander (or so it seems) who is kidnapped, taken to the headquarters of one of the corporations and then we see what it means to be forcibly conscripted. The unfortunate man is placed inside a large machine called Leonardo, whose function is to graft the poor man with biomechanical limbs and make his mind controllable.

Returning outside, to the spot where the man was abducted, we find him there, ready to carry out the orders that will be imposed on him.

The presentation of Syndicate is graphically very good, the design of vehicles and environments very apt, the animations simply impeccable. Excellent also the use of colours, sepia for the exteriors, cold and metallic blue for the interiors. The music then contributes to create the sense of oppression and tyranny in which the world population is forced to live.

A truly spectacular and atmospheric introduction to an absolutely magnificent title.

AGONY (BONUS)

Agony

Why bonus? Well, because Agony has no animated intro. So what's it doing here? The answer is very simple: music.

A wonderful musical theme of over four minutes, composed by Tim Wright, accompanied by a beautiful illustration by Frank Sauer (as well as all the beautiful images that introduce each level of the game), introduces us to the fantastic world of Agony. Piano and strings create a haunting, intimate and sublime melody.

I don't want to write any more, I wouldn't be able to describe such beauty, but I'm sure that all of you Amighisti brothers and sisters know exactly what I'm talking about. But if not, catching up is essential.

Pure art and wonder, if you'll pardon the pun.

And here we are at the end of this list, I have chosen these ten wonderful intros but there are many "excluded" ones that deserve to be mentioned, maybe in a future second list of the most spectacular Amiga intros.

And which are your favourites?

Visit us on our social channels and tell us your list!

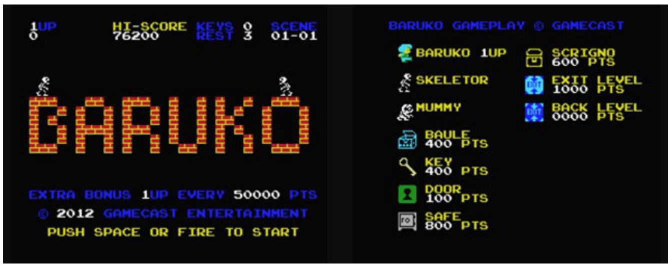
Cheers to you all and **AMIGA FOREVER!**

Talent Scout

by Carlo Nithaiah Del Mar Pirazzini

A new column is born to discover and talk about the large group of developers, graphic designers, musicians who are doing so many great works for our retro-computers/ retro-consoles. Today we present Andrea Gasparrini and his GAMECAST Entertainment.

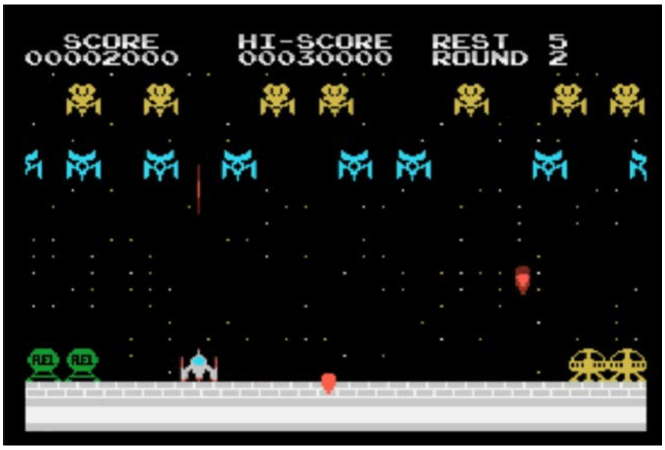
Let me introduce myself, my name is Andrea Gasparrini and I'm a passionate independent developer and retrocomputer enthusiast.
I have always followed the MSX/MSX2 standard, from the 80's until today, first on MSX 1 and MSX 2 and later on MSX 2 Turbo - R and MSX 2+.
From my great passion for video games I became a small indie developer for MSX/MSX2 and I work with Basic, Turbo Basic, Nestor Basic and in Assembler Zilog Z80.
But at home I also own some 8 Bit Commodore machines and currently three MSX 1.



BARUKO – MSX1: for the MSXDEV COMPO 2012

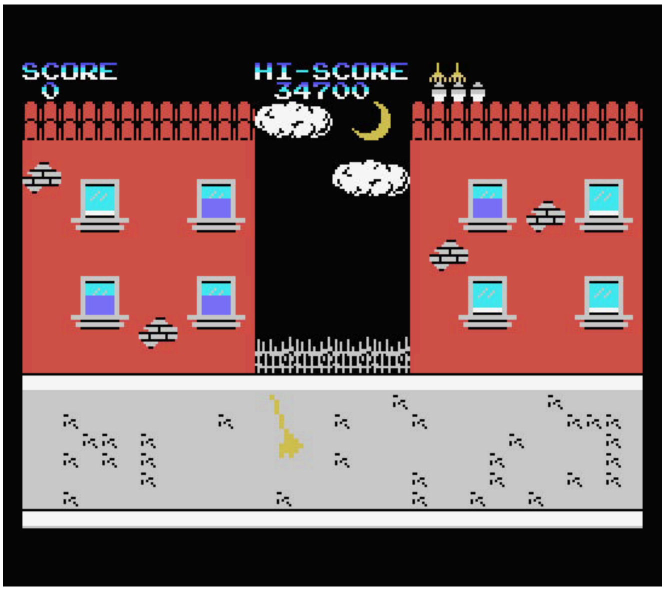
In the past I have participated in many contests of the MSX scene as MSXDEV and also in the MSXDEV COMPO. The name of my team is GameCast Entertainment, even if previously you can find me with the name AGSoftware (Andrea Gasparrini Software).

Also this 2022 I will participate in MSXDEV '22 with my new space video game called DEFENCE, although I think I can squeeze in a few more titles.



DEFENCE game for MSX competing at MSXDEV '22

In 2021 I participated with the title PIPPO LA SCOPA FICCANASO (in English: Goofy the Nosy Broom).



PIPPO LA SCOPA FICCANASO – MSXDev '21

Among the many jobs I'm doing and have done, in 2017 I attempted the publication of an online pdf review I called DATAPRINT Magazine.
You can find it here: https://dataprint.altervista.org/

My passion as a programmer was born thanks to the LIST magazines of the 80's published by Edicomp that made me fall in love with Basic.

Greetings and thanks again.
Andrea Gasparrini

Useful links

Game Cast Entertainment:
http://gamecast.altervista.org/index.php

MsxDev '22:
http://www.msxdev.org/

DATAPRINT Magazine:
https://dataprint.altervista.org/

NEW GAME

RANDOOM

Year: 2021
Code: Dozznar of Picaro Games
Music: Narcisound of Onslaught
Graphics: Toni Galvez of Batman Group
Publisher: Picaro Games
Genre: Platform
Platform: Commodore 64

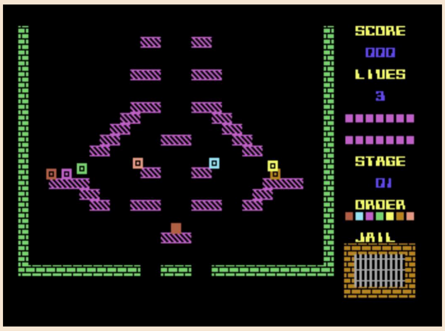
Is it possible that a game made up almost exclusively of squares as big as sliders could make sense? And above all be fun? If we talk about Commodore 64 and Randoom the answer is YES. We often say that this rebirth of eight (and sixteen) bit platforms is perhaps the biggest thing that could happen to the gaming market. The titles that are coming out for our retro gaming machines are more and more incredible, but even more are the directions they take.

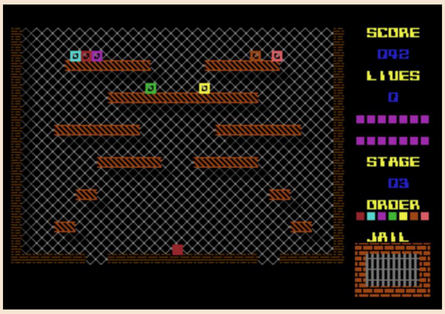
The game we are going to talk about today chooses to take a literally random direction, as can be seen from the name itself, and makes this feature its strong point. The concept can be summarized in a few lines: we will find ourselves chasing cubes through 31 game screens, making sure to take only the one of the required color, under penalty of losing a life. Written like this maybe doesn't mean so much, but even more unsettling is when you start playing a Oscar-winning music is counterbalanced by graphics that to define simplistic is an understatement. I stood still for about thirty seconds on a fixed screen to see colored squares move: I was looking for my character. I decide to move the Joystick and I see another square moving that was stationary in the middle of the screen: I fall into a hole, I go out from the top of the screen, I fall again, I collide with a square and die. It starts all over again, I fall back, I hurt a cube and die. Here I am again at the beginning, but this time I press the fire button and see my cube jump. I stand still and, analyzing the game screen, I understand that I must take the escaping cubes in a precise color order that appears in the section dedicated to game info.
In that moment a historical name comes to mind: Bubble Bobble. Indeed, the glance is that: we have the main game screen on the center and information on the right (score, lives and so on). Even the simplified dynamic can fit: we move on platforms, we jump and we have to capture enemies... only the bubbles are missing. Nothing could be more wrong my dear readers, because once you have overcome the impact with the basic graphics, you realize a detail that will be the strength of the title in question: the enemy cubes do not follow a predetermined trajectory. I quickly realized that each time I died the level would restart, but cubes always moved differently. Thus, blows up the old theory that "to learn the enemies' pattern for proceeding in game": here we literally have to improvise.

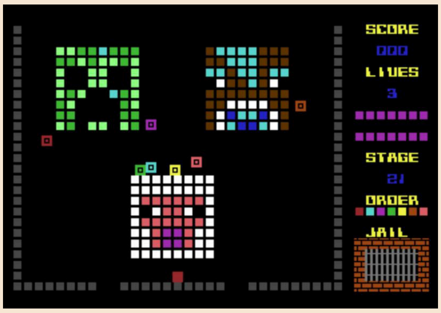
We will find ourselves chasing, fleeing, jumping, and always looking for our target in a different way. It's quick to get carried away and die less often but there is no rule, it depends on how the colored cubes will move and based on this, the same level can be easy or a real hell. Randomness, here is the absolute stroke of genius that will make us literally stick to this game for a long time. If we manage to capture enemies in quick sequence, we will have bonus points and, once we reach 100, an extra life that never hurts.

I want to find a flaw: the first and the penultimate cube to catch are chromatically very similar and sometimes I found myself capturing the wrong one. But apart from this we have in our hands, at no cost, a crazy game, an arcade platform that confirms us how a good and brilliant idea is the basis for a great game. In the end, the simple graphics are good for the game itself, while the audio track is truly killer, one of the best produced by SID in recent years. I close my eyes for a moment and imagine Randoom in an arcade bar in 1986, in a great dedicated cabinet: it would have been a success, perhaps with the possibility of having a 2 player game!

by **Mic the Biker Novarina**







OUR FINAL SCORE

» Gameplay 88%

A few plays to understand what to do and how to do it, at first dying is easy but you quickly learn how to proceed.

» Longevity 98%

Each level will never be played in the same way; the randomness factor makes this title potentially infinite. You will not get away from it anymore!

NEW GAME

RANDOMINER

Aear: 2022
Deeveloper: Thomas Ilg
Genre: Puzzle
Platform: Atari ST
Website: https://www.hd-videofilm.com/randominer/

The game system is the same as the famous Windows "Minefield" but brought in a isometric view and with several additions.

You have to click to open a tile. The tile number will tell us how many mines are present around us.
We'll have to use logic, brains and often a high dose of luck to determine where the mines are hiding.

There are a few more aspects to this, such as extra bonus points for quick plays that put a little "salt" to the challenge.

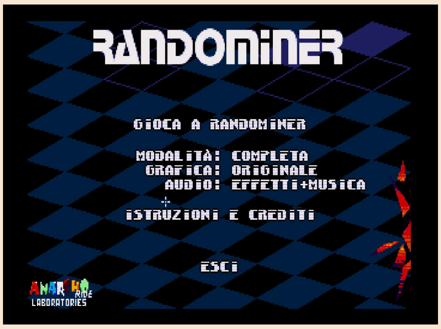
The game runs on all machines with at least 1 mb of Ram. It loads very fast and runs without problems even in emulation.

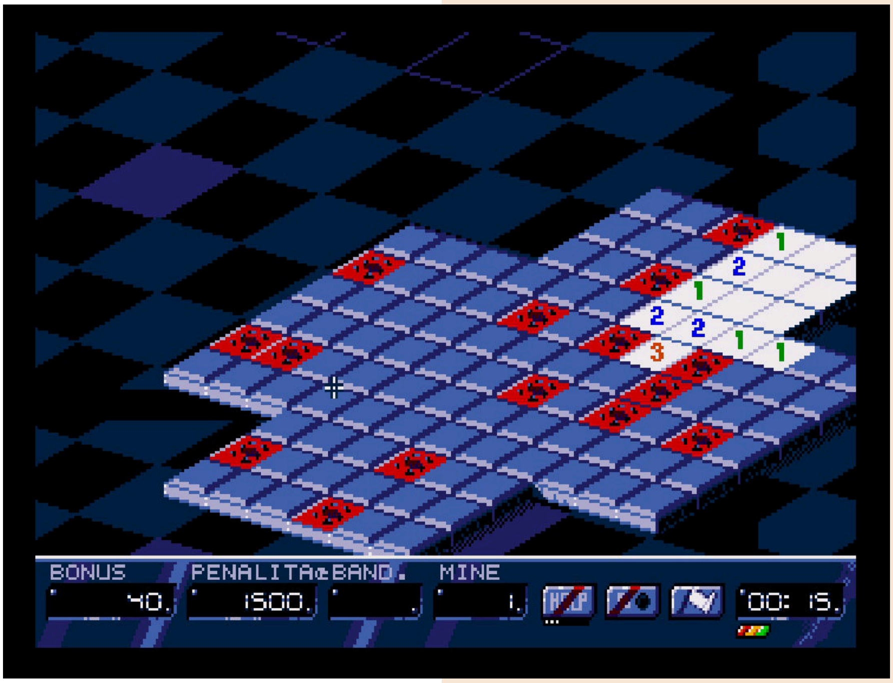
It has been localized in many languages, among them we also have the Italian language edited by our dear friend **Filippo Santellocco** of the **Atari World Italia** group.

It's a well developed and solid game, the classic title to load to spend a few pleasant minutes in the breaks from real life.

Highly recommended.

by **Giampaolo Moraschi**





OUR FINAL SCORE

» Gameplay 80%
One of the most played puzzle games in history in a modern twist.

» Longevity 80%
You will load it up every now and then, in between breaks. I found it well done and relaxing.

NEW GAME

GREEN BERET

Year: 2022
Editor: Konami
Genre: Shoot'em up
Platform: Amiga
Website: https://danteretrodev.itch.io/greenberet

Another 80's classic comes to the Amiga. It took its time, it had to wait for the arrival of the Scorpion Engine, but it's finally here.

These last two years have been full of conversions of historical titles on Amiga, such as Yie-Ar Kung Fu or Metal Gear, and it's time to dust off another arcade classic, Green Beret or, as it was known in the USA, Rush'n Attack.

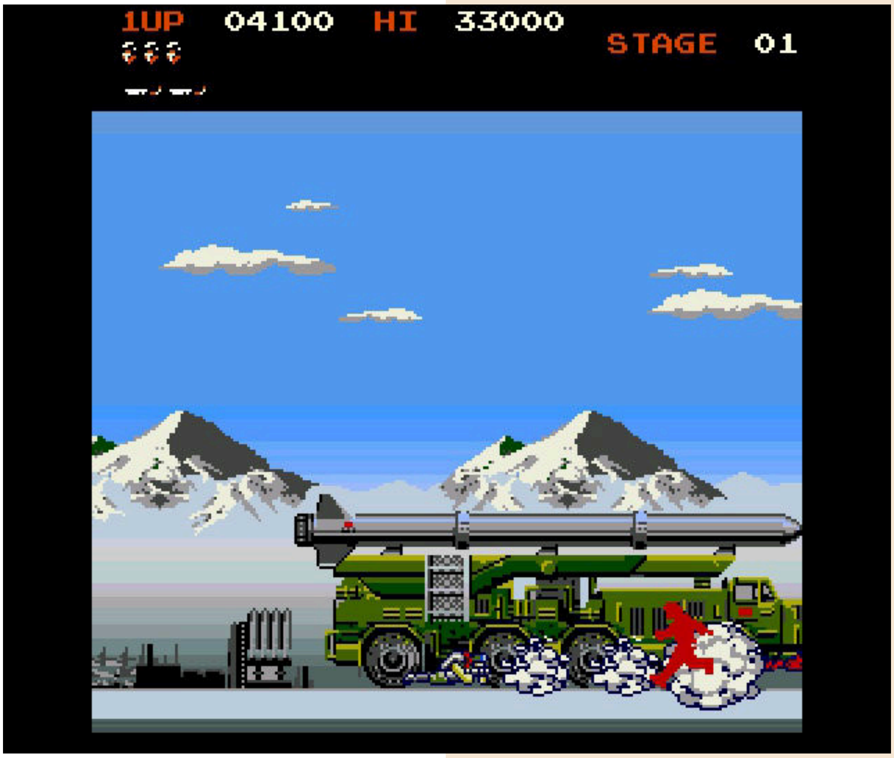
The game was created in the historical moment when the world was divided in two (more or less like now, but in a more decisive way).

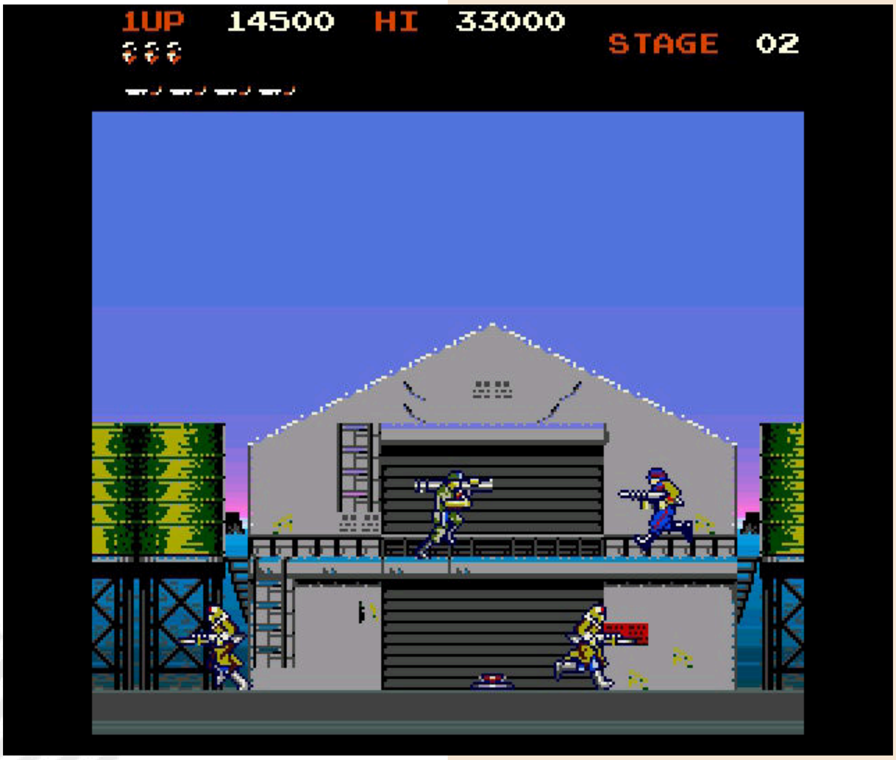
We will play as a heroic soldier of the Green Beret Corps who has the task of freeing hostages in some prison camps of a phantom "rogue" state.

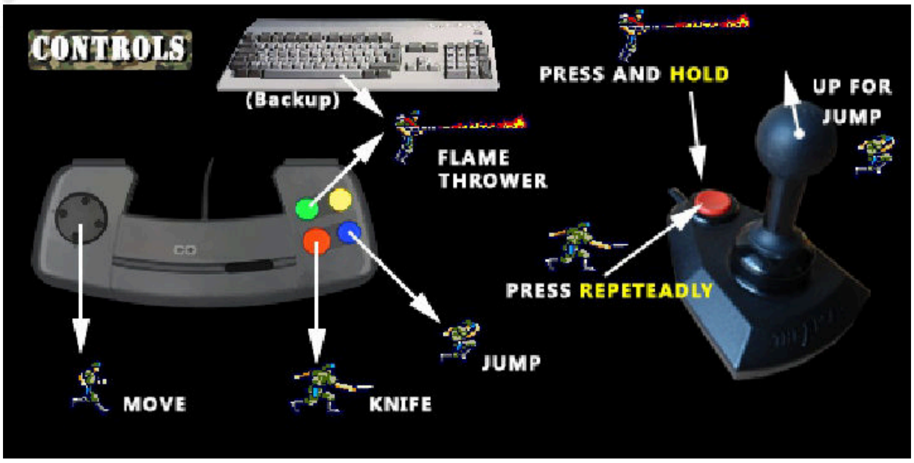
To do all this we will be armed with an incredible sharp dagger and only three lives. Convenient right? Because all the rest of the weapons we'll have to get them ourselves with our strength alone.

Enemies will obviously try to skin us in every possible way. We'll need steady nerves and a strong memory to make it to the end of the game.

It is a dynamic and difficult game like the best arcade of the fabulous eighties. In fact, at the time, it was an incredible success and there were many conversions for home computers.







OUR FINAL SCORE

» Gameplay 85%

It has it all. Weapons, levels, enemies, and ...swearing!

» Longevity 85%

It has the feel of the arcade game and that way of involving you more and more, game after game.

The best versions were those for Commodore 64, followed by those for ZX Spectrum while on consoles the NES version was a great success thanks to the addition of new levels and some small bonuses.

But on Amiga? There was nothing, or there was Fire Force but it wasn't the original.

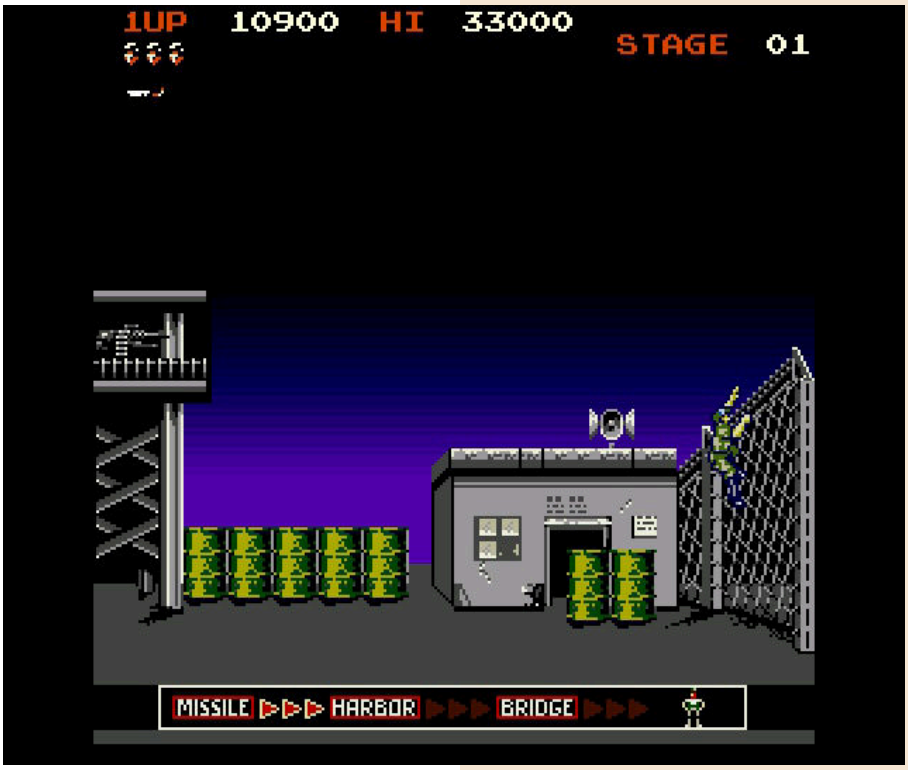
In 2022 all this is solved with a nice version made with the Scorpion Engine. There are the 4 levels of the original game with all the end-level bosses, all the enemies, the three secondary weapons of the arcade and can also be played with the two-button gamepad in addition to the canonical joystick.

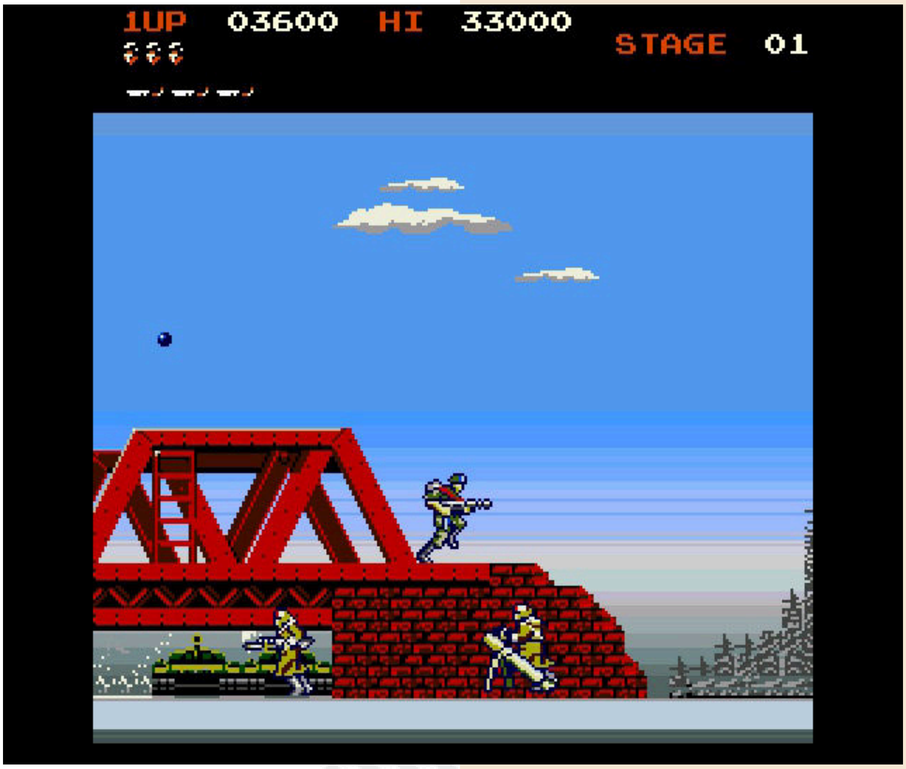
The game has a nice soundtrack redone from the NES version and runs on all machines with 1 Mb of Ram.

A title that looks identical to the original, as is identical to the gameplay and approach to the game.

Green Beret may not appeal to new generations, but all those who loved it in the arcade will appreciate it very much.

by **Roberto Del Mar Pirazzini**



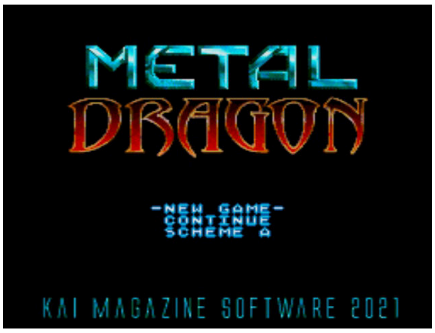


NEW GAME

METAL DRAGON

Year: 2022
Developer: KAI MAGAZINE SOFTWARE
Genre: Shoot'em up
Platform: Sega MegaDrive
Website: https://kai-magazine-software.fwscart.com/

The Megadrive is an incredible console! It is enjoying a second youth thanks to a large group of developers, graphic designers and programmers. In recent years we have seen the release of titles in massive dose and almost all of good workmanship.


METAL DRAGON

-NEW GAME-
CONTINUE
SCHEME A

KAI MAGAZINE SOFTWARE 2021

This Metal Dragon is a title created by the collective KAI MAGAZINE SOFTWARE, who had previously released it on MSX V9990. The team is led by Oscar Kenneth Albero and is based in Spain, a land that has given us so much on the retrogaming front.

It's a great trip through time. A journey through the mythical titles of the first period of the console that takes us back to a "macho" era and recalls titles like Mercs or Dynamite Duke. Titles that turned us into "tough guys" who alone faced entire armies to liberate the world.

It is clear from the beginning that this title for Megadrive is an exercise in nostalgia, but it is also a parody of other titles, especially the Metal Gear series, to which it mercilessly launches many provocations.

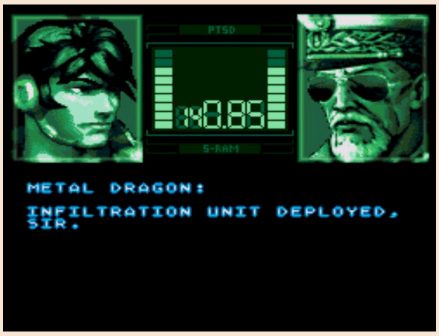
Its development is hopelessly reminiscent of the aforementioned Mercs or Commando.

Leading the hero who gives his name to the game, we'll have to advance through the level while hundreds of enemy soldiers will shoot at us with any type of weapon.

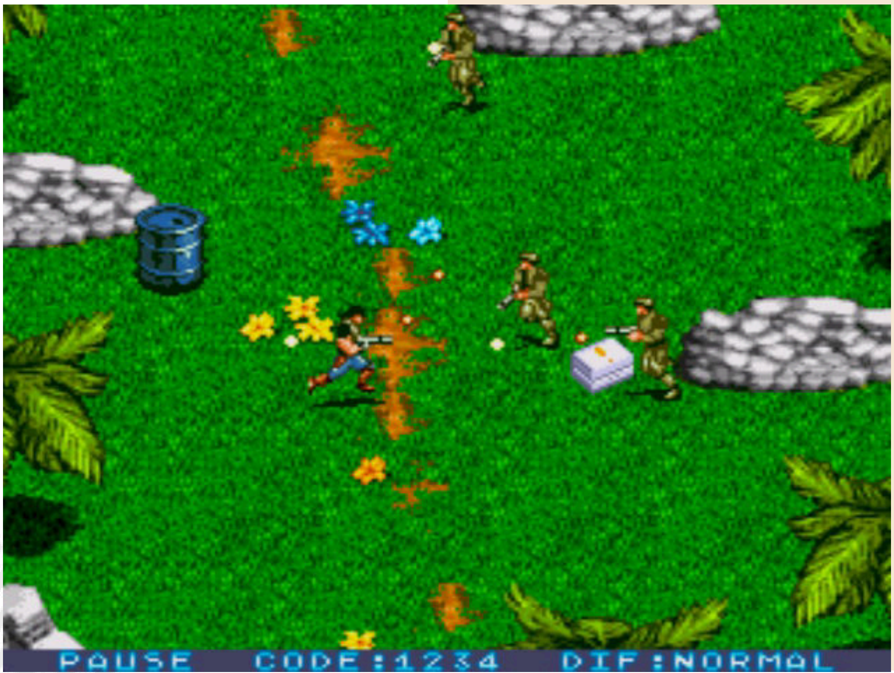
Obviously we have to shoot them to eliminate them, but the important thing is to advance to the end of the screen to move to the next area. In other words: except in duels against final bosses, it is not strictly necessary to eliminate anybody... Although the situation constantly "invites" us to shoot.

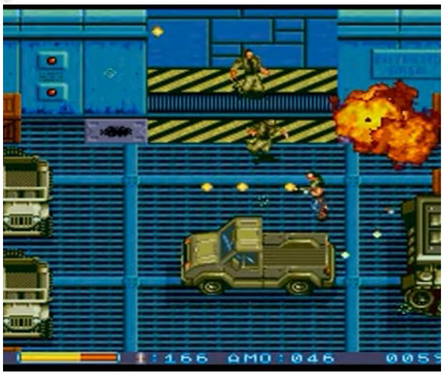
You'll start with a weapon with unlimited ammo, but with very low damage. Fortunately, eliminating some enemies we'll be able to get a beautiful and very powerful arsenal that ranges from the flamethrower

MEGA DRIVE
METAL DRAGON
KAI MAGAZINE SOFTWARE 2021



METAL DRAGON:
INFILTRATION UNIT DEPLOYED, SIR.


PAUSE CODE:1234 DIF:NORMAL



to the machine gun, from the bazooka to the sniper rifle.

All of these weapons have limited ammunition and the same goes for grenades, whose quantity will only increase by finding some boxes in the levels. Grenades are essential to eliminate some particular enemies, such as robots or tanks that appear.

Reserving our best weapons for the most difficult enemies is a very good idea, although playing at normal difficulty you will not have problems storing ammunition in general (big flaw).

We'll be able to pick up first aid kits found in levels to restore our body damage, though it's always possible to withstand several hits without dying.

We don't have a number of lives, but a health bar that, if minimized, will lead to Game Over. A convenient password system and the many continue will allow us to continue the adventure without too much trouble and without a real challenge. In this case I would have preferred less continue to make the longevity higher.

The game is still very long and requires commitment and a lot of memory. There are four difficulty levels but only the Hardcore level represents the real challenge.

The game also features two endings. One very catastrophic and one with a happy ending.

It lacks the possibility to play in duplicate and I must admit that it is a pity for this kind of games. In Mercs it was a wonderful possibility.

Graphically it is very interesting. Excellent use of color and design of enemies as the quality of detail of the backgrounds. Every now and then we see some flickering of the sprites but does not compromise the vision of the game.

Soundtrack with excellent effects digitized but the accompanying music a bit 'flat.

Strong piece is the humor that permeates the dialogues of the game, which as we said, hits hard Metal Gear and makes fun of several action movies. A bit like that masterpiece of UnMetal (reviewed in issue 33 of RMW).

Summing up, I'd say Metal Dragon is a good action game that does justice to the golden age of the Megadrive. It's a decent challenge for our reflexes and has a very deserving technical section, especially if we take into account the very small group that "gave birth" to the title.

I would have liked a little more length and a higher difficulty (only the hardcore level is difficult).

by **Carlo N. Del Mar Pirazzini**

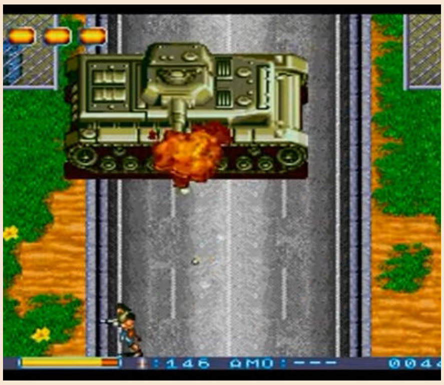
OUR FINAL SCORE

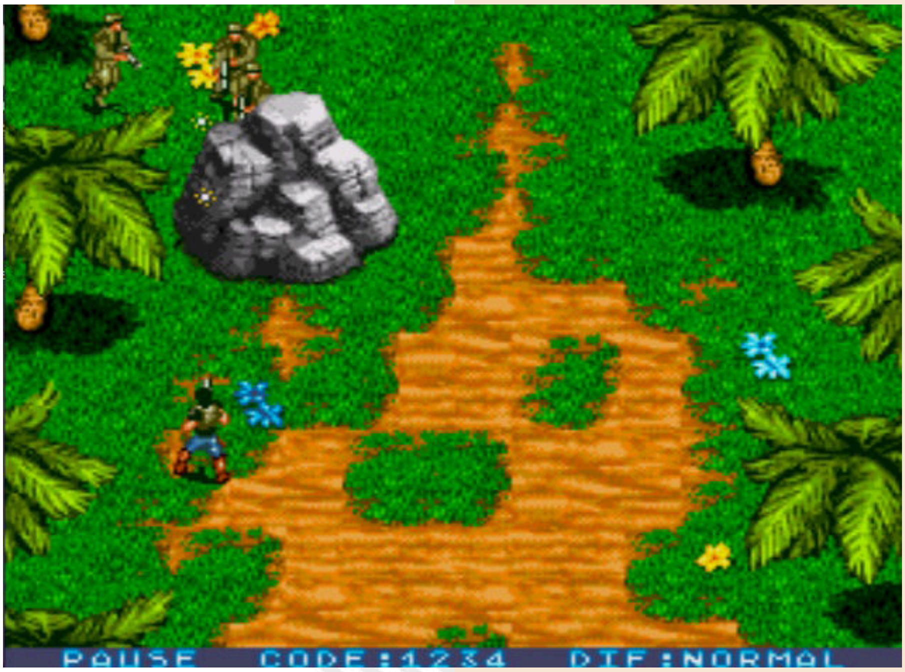
» Gameplay 80%

This is a remarkable exercise in nostalgia, humor, and a passion for the classic "guerrilla warriors" of the past. It lacks some aspects, but the "journey" is very enjoyable. The action and firepower are there in Metal Dragon.

» Longevity 70%

There are some gaps in the difficulty curve. The difficulty levels aren't that diverse and the endless continues/ password system will have you ending it way too early as well. The two endings are worth watching though.





STAR OCEAN

Year: 1996

Editor: Enix/Tri-Ace

Genre: Action/GDR

Platform: Super Nintendo

The Super Nintendo is known to boast a rather large and exceptional library of role-playing games, but some of these little masterpieces have never appeared outside Japan.

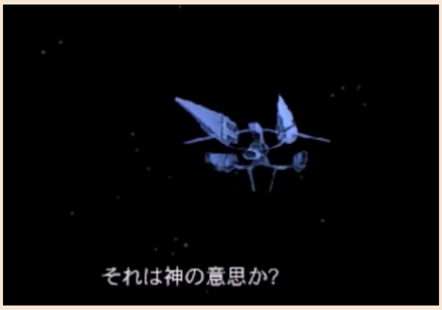
Star Ocean is at the top of the list of unknown titles, but thanks to a group of 'translators', a patch has been created that allows the title to be played in English and/or Spanish.

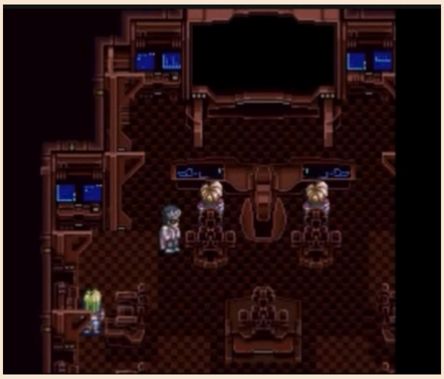
Star Ocean's biggest distinction comes from the fact that it is the culmination of the work of several developers who originally worked on the successful Tales of Phantasia, which explains why the game sports so many similar features and styles.

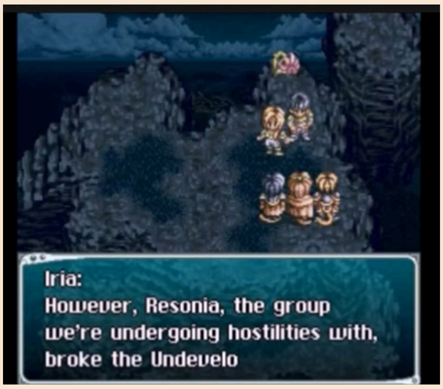
That said, it's worth noting that Star Ocean is a distinctive role-playing experience that features a truly unique battle system and some of the most cutting-edge technology available for Nintendo's 16-bit console at the time of its release. It is also the reason why the translated game took 26 years to come out and be compatible on an emulator. But I assure you that all this time spent waiting will pay off. Visually, this is one of the most beautiful Super Nintendo games ever created. The original game used a special graphics chip inside the cartridge, called the S-DD1 that allowed developers to use complicated graphics sets during gameplay while still fitting within the ROM size limits. This was one of the reasons why Star Ocean was so difficult to emulate and for a long time required numerous graphics add-ons to play.

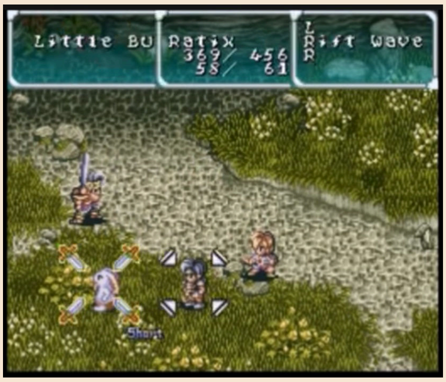
Each area of the game features a highly detailed and lush environment, a level of detail rarely seen on a 16-bit title. Each area is strongly characterised and has its own unique look and feel and we rarely see anything repeated twice, which has the effect of making the game incredibly vast. It's clear that the developers have put a lot of time and energy into bringing such graphics to life and have spared no expense in doing so. Without a shadow of a doubt Star Ocean is one of the most graphically beautiful titles for this console and perhaps the perfect example of visual capabilities.

As if the graphics weren't enough, the soundtrack never fails to impress. The music is reminiscent of the epic Final Fantasy titles, but with a more melodic touch that sets it apart from other Enix productions of the period. Again, the musical variety is impressive and no music track is repeated within the title. A peculiarity of the sound is the use of audio samples. During battle, we often hear shouting and banging out dialogue. By modern game standards this sounds nothing spectacular, but at the time there was


それは神の意思か?




Iria:
However, Resonia, the group we're undergoing hostilities with, broke the Undevelo



nothing like that before the Playstation.

But the strong point of this game is its unique real-time combat system that mixes the action combat system with a more traditional turn-based combat engine.

Real-time attacks can be performed, but instead of wielding a sword, we can select an attack command. We'll be able to target specific enemies, as well as use items and special magic attacks that can be assigned to the back buttons of our pad. You control one character at a time, but we will be able to scroll through the available characters and change the one you are using at any time during the battle. The other PGs will be controlled by the game's AI.

As in most classic role-playing games, in Star Ocean you'll spend a lot of time exploring the game areas and occasionally launching into some side quests to gain experience. These additional quests, which the game calls "Private Actions", are not crucial to the completion of the story, but they add a lot of detail and depth to the plot, making the title less linear.

It's easy to see why so many fans of role-playing games on the Super Nintendo have held this title in such high regard for so many years. It's also easy to see why it was one of the first games to be translated into English by the Dejap team. Now that the game's graphics have been fixed and decompressed, there's no reason why gamers shouldn't patch the rom and give this gem a try.

Star Ocean is easily one of the best role-playing games available for the 16-bit console and ranks right up there with The Legend of Zelda: A Link to the Past.

Over the years, a nice PSP version has also been released, but it is less attractive than this version.

Try it.

by **Takahiro Yoshioka** and **Carlo N. Del Mar Pirazzini**

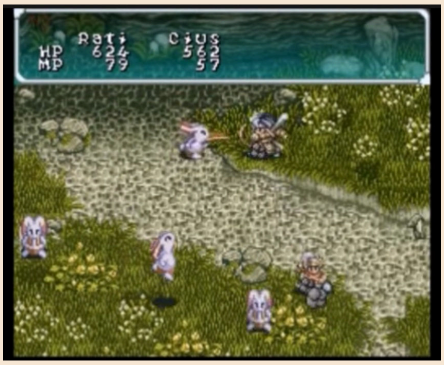
OUR FINAL SCORE

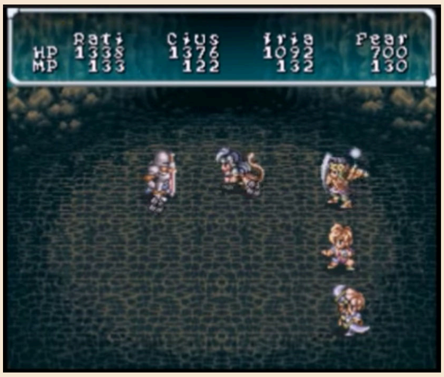
» Gameplay 98%

Incredible control system and a mix of pure action and turn-based combat. Lethal but incredible.

» Longevity 95%

A long-lived and vast title to explore and discover. Beautiful side quests.





NEW GAME

ATTACK OF THE PETSCII ROBOTS

Year: 2021/2022
Editor: The 8 Bit Guy
Genre: Action
Platform: C64/VIC 20/PET/ Amiga/APPLE II/PSP
Reviewed version: C64
Website: https:// www.the8bitguy.com/

Robots have taken control of human settlements on a number of planets and it is up to us, fearless heroes, to infiltrate and destroy all the rebel machines.

To do all this, however, will require a bit of strategy and our task will be to explore the game environment and try to locate weapons, health kits and cards to access certain areas.

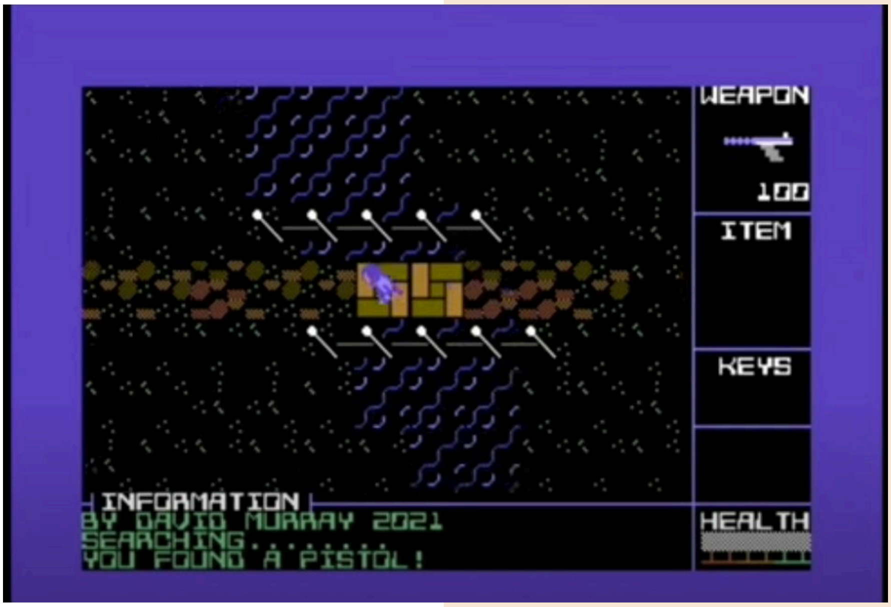
Each robot presents a different level of challenge: some are very weak and can be eliminated with the basic weapon, while others are more difficult and cunning, and to take them out, we will have to be cunning and use more powerful weapons.

There are 9 game maps available that require great tactics and study and an excellent degree of challenge. Beware, however, that exploration is "lethal". There is no save system or password and if we get to the death, we will have to start again from the beginning.

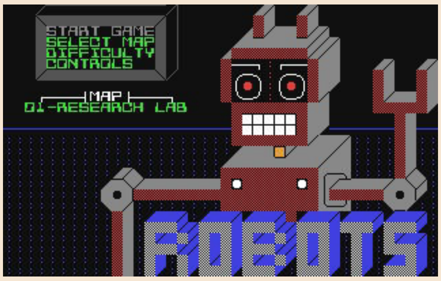
Attack of the PETSCII Robots has a unique look and feel compared to current games. Similar to what we found in mid-80s titles... a nostalgic appeal

.

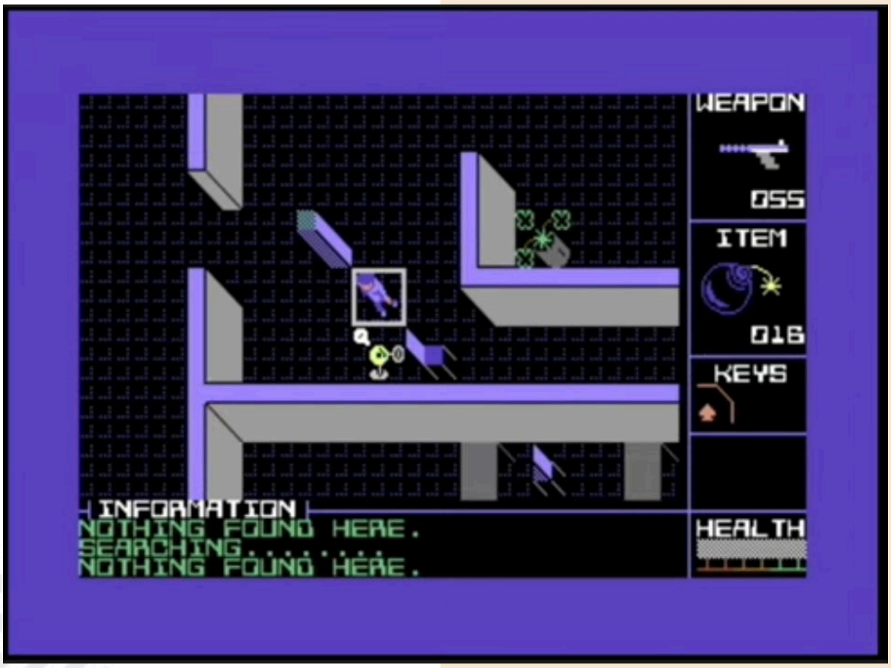
Its graphics do an excellent job (you can play the title in three graphic modes: Classic, PET or Colour PETSCII). Well animated and undoubtedly impressive for this title. Scrolling is a bit jerky, but with a limited field of view there's a bit of uniformity and it doesn't bother you too much.



BY DAVID MURRAY 2021
SEARCHING...
YOU FOUND A PISTOL!



Having more variety among the robots



NOTHING FOUND HERE.
SEARCHING...
NOTHING FOUND HERE.

would have improved the game more by increasing the challenge.

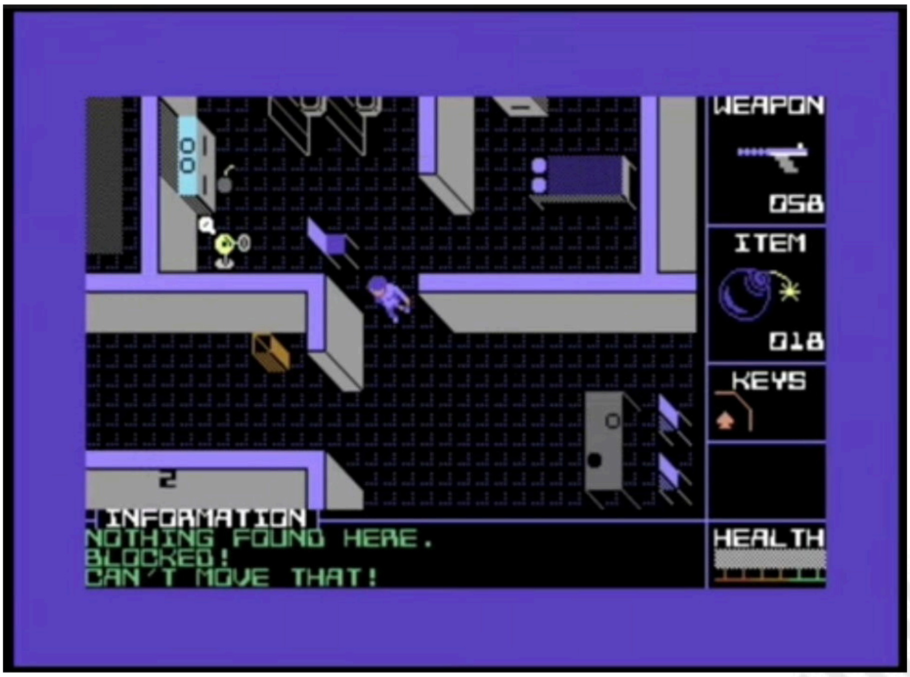
There are several ways to take out the robots and searching for objects during the adventure is fun.

The real problem is the lack of a save system or checkpoint or password. A detriment to the game, because each level is played exactly the same way every time, and when you die you are asked to repeat the exact same previous process.

This deficit greatly lowers the final longevity.

It is still a good title.

by **Roberto Del Mar Pirazzini**



NOTHING FOUND HERE.
BLOCKED!
CAN'T MOVE THAT!

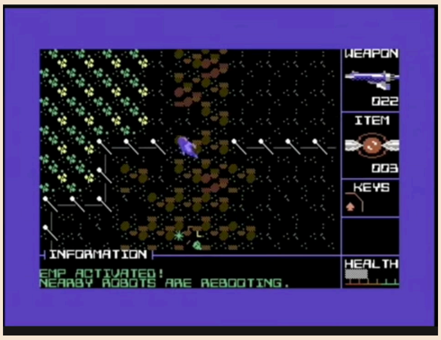
OUR FINAL SCORE

» Gameplay 80%

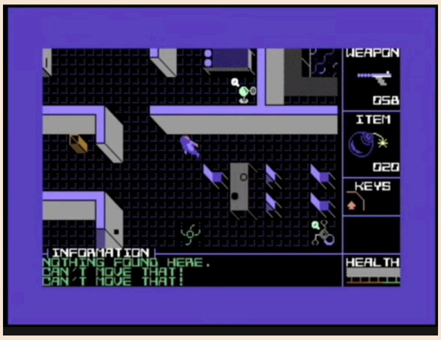
Lots of items to look for, 9 levels and a number of interesting and different ways to eliminate robots.

» Longevity 60%

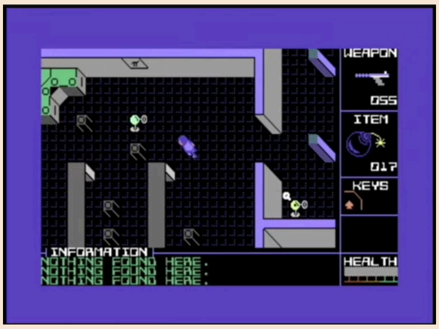
Only three varieties of enemies and the lack of a save system bring down the final longevity heavily. Pity indeed.



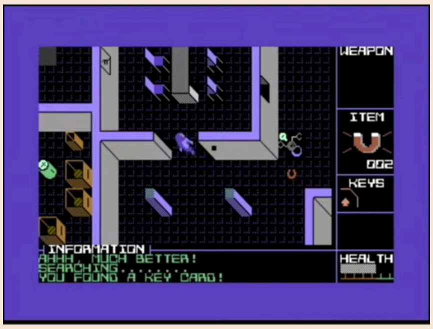
EMP ACTIVATED!
NEARBY ROBOTS ARE REBOOTING.



NOTHING FOUND HERE.
CAN'T MOVE THAT!
CAN'T MOVE THAT!



NOTHING FOUND HERE.
NOTHING FOUND HERE.
NOTHING FOUND HERE.



AHHH, MUCH BETTER!
SEARCHING.......
YOU FOUND A KEY CARD!

KLONOA 2
LUNATEA'S VEIL

Year: 2001

Editor: Namco

Genre: Platform

Platform: Playstation 2

It's nice to go to Nith's house, in the chaos of his attic, among disassembled computers, game consoles, I don't know how many Commodore 64 (I have only 6! - Nith's note) sometimes you get lost.

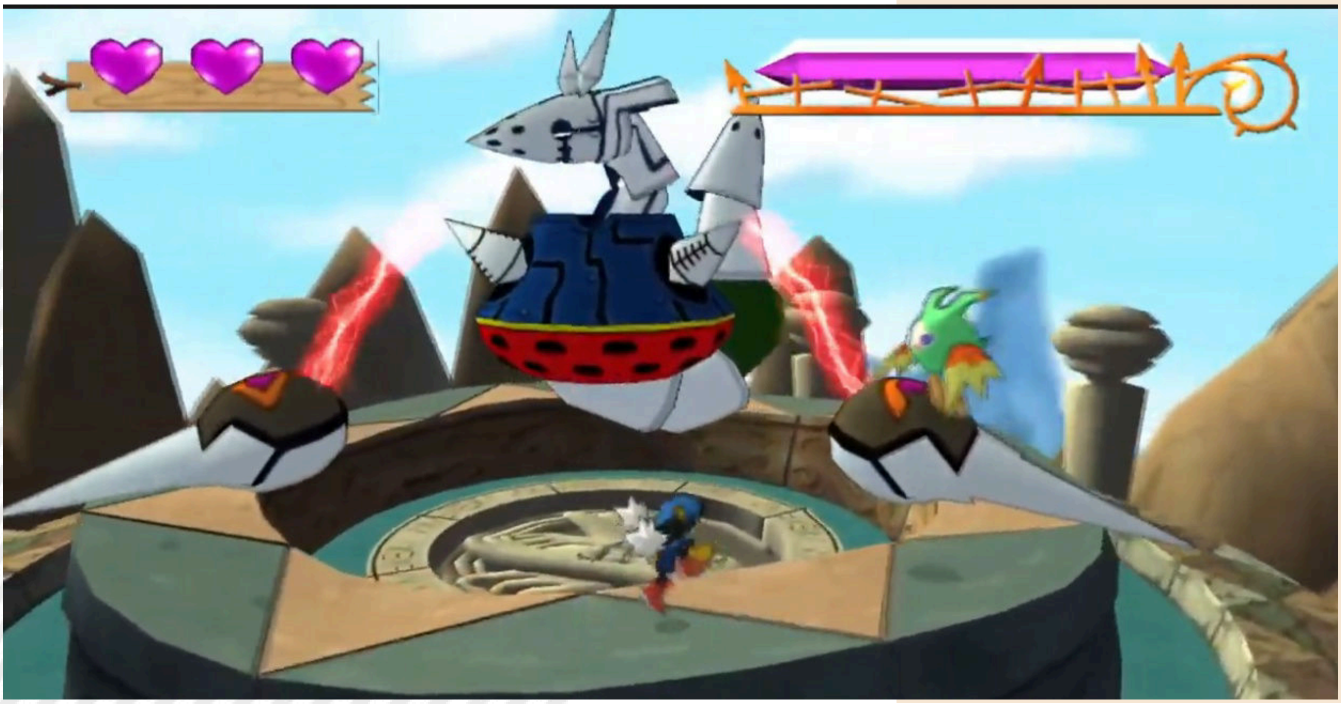
In my period of "reclusion" due to Covid, I spent some time looking in my spare time as a student at some interesting titles and here, among several titles and covers, I see this Klonoa 2.

Why am I drawn to it? I remember that when I was little I played it on my father's Playstation, but this is the sequel and I wanted to try it.

I turn on Nith's Play 2 and the magic begins.

The first Klonoa had really got me. Maybe it was because of the cute and funny main character and his cartoon world, but I really got into it and finished it.

The story of Klonoa 2 follows the floppy-eared protagonist through his adventures in the dream world of Lunatea. This enchanted world is kept in balance through the four bells of harmony, located in the four different realms. But a terrible evil force is



planning to bring a fifth bell, that of Sorrow, which will make the priestesses sick and numerous monsters appear throughout the realm. It's up to Klonoa and her two new companions - Lolo a young sorceress and Popka a strange glass-eyed character - to prevent this evil plot from coming to fruition.

Okay the corny Japanese New Age story I've brought back, now let's talk about the game itself.
It's not an original title. It's a platform game that takes what we saw in the first chapter and adds a couple of new dynamics. As in the first, we'll be able to grab Yoshi-style enemies using our power ring and use them to propel us higher than normal jumping, or use them as projectiles to throw at other bad guys or to activate switches or blocks in the way. There are several new enemies in Klonoa 2 that grant us special abilities once captured, such as the Kiton that acts as a helicopter for a short time, or the boomie that becomes an explosive.

The game always keeps a good balance between simple puzzles and jumps from one platform to another. In short, it's always fun.

Too bad about its devastating lack of difficulty. The game seems calibrated for very young children and in less than 6 hours you will complete it. Too bad really because the pace of the game is pleasant and you wish it would last at least twice as long.

Nice graphics, colorful and fast. Nice sound... Simple commands. But holy crap! This lack of difficulty has left me stunned.

Too bad really. A good platformer but nothing more.

by **Ingrid Poggiali**

OUR FINAL SCORE

» Gameplay 80%
Fun mechanics taken from the first chapter for Play1 and new additions.

» Longevity 40%
Calibrated for a really young audience. It ends and leaves us desperate and wanting another piece of gameplay.

TWINKLE TALE

Year: 1992

Editor: Wonder Amusement Studio – ZAP Corporation

Genre: Run and Gun

Platform: Sega Megadrive

Retromagazine World, the pleasure of discovery. Our journey among the unknown titles for some platforms continues. After viewing some shoot em ups for NES today is the turn of the Sega Megadrive.

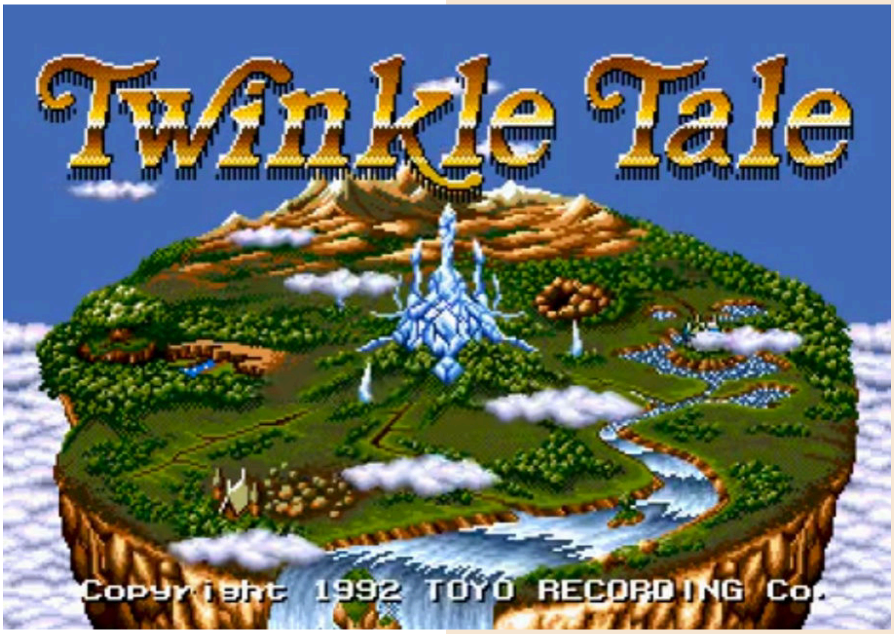
Twinkle Tale is a run and gun with top-down view released in 1992 exclusively on the Japanese market. It stands out from other titles of its kind for its unique setting: it combines a gameplay reminiscent of Mercs or Knightmare by Konami (have you read the review on the last issue of the Amiga version? - Nith's note) with a nice horror-kawai theme that reminds us of the Cotton series.

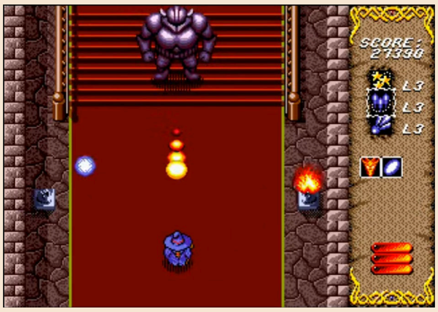
We face each level mowing down everything in sight and absorbing every power up possible with our character, the cute little witch named Saria, complete with a nice magic stick and pointy hat.

The enemies are a mix between the classic fantasy monsters and those of Japanese mythology and each level has characteristic and above all, fleets of them!!!

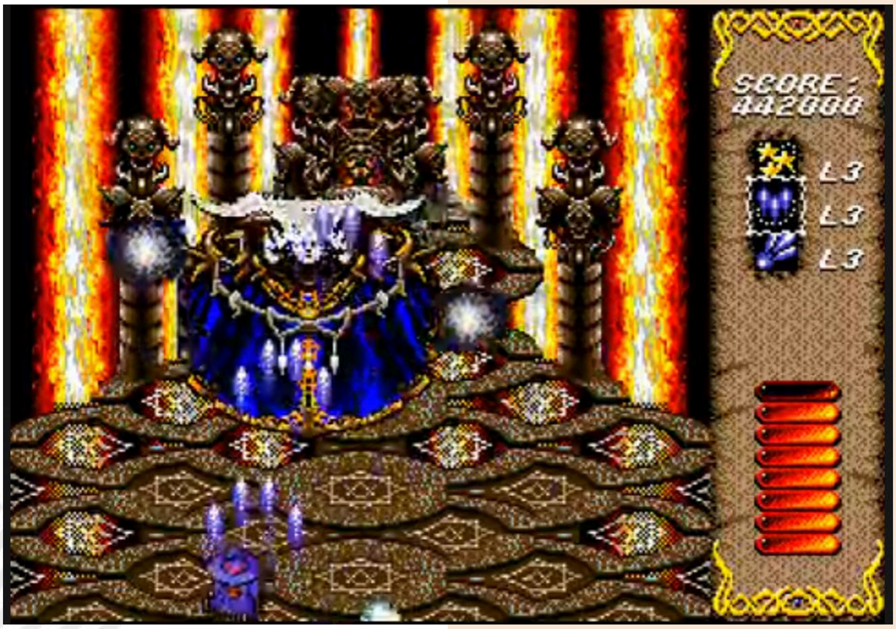
The title screen shows a floating island and the introductory screen that explains the story, made to look like a picture book, which sets the perfect tone for the game.

First of all, Twinkle Tale is a fast-paced game. Saria finds herself immediately attacked by a number of fierce and fast enemies. Luckily for her (and us), she is equipped with a standard solo weapon and a number of secondary weapons that are released by the enemies she faces. The C button on our pad will toggle between weapons, and the B button will fire the currently selected weapon.





The Shooting Star fires a wide, high-powered stream directly in front of



us, the Diamond Arrow fires in multiple directions and simultaneously, and the Silver Comet is a slightly lesser fire but has enemy tracking capabilities. This makes the title very varied and gives a decent amount of variety compared to other similar titles. There isn't one weapon better than the other, but we'll find that we'll generally get better results with specific weapons against certain enemies. This mechanic was later used for some levels in titles such as Radiant Silvergun and Neo Contra. Each weapon can be upgraded by finding stars hidden inside some treasure chests, which will increase the firepower up to three times. If you get hit, before dying and going into game over, the energy bar will lower thanks to the number of upgrades we'll get with the weapon.

Also present are the essential bombs, and they are of two types: the first has a search capacity while the second creates a full-screen Krakatoa. Both are triggered with the A key.

There are nine levels in total, all of which are quite challenging. You can change the initial difficulty in the game options and you can also activate a little trick that allows us to jump to the desired level (in the storybook home screen we see after pressing start, we press Up, Left, A and C and then start and we can select the level with the D-pad).

The boss fights are splendid. The boss of each section is approached as in a classic vertical shooter, with the character only able to shoot upwards. Each boss is huge, with movement patterns and fire patterns that bring to mind what you would expect from a shooter. If you're a fan of this genre it's a real treat.

Another thing that really stands out in Twinkle Tale is the level of variety in its stages, which constantly involves more than just moving in the appropriate direction and blowing things up. A particularly good example of this fact is the third level, where you'll have to traverse a series of stone walkways in the sky. Unlike most similar games, where you can walk along the edge without worrying too much about falling, here if you're not careful you'll be hurtling towards game over. Timing, concentration and dexterity are required. The game was made in the middle years of the Megadrive's life, but it looks like a title for the next generation of games for this console. The graphics, although still grainy compared to a Snes title, is among the best on this platform. Colorful, with animated and well-detailed sprites and fantastically designed backgrounds. Many levels show glimpses of the world and are incredible.

The Boss design is beautiful. Exceptionally well designed and impeccably animated.

Nice audio support but not incisive, in the end blasting enemies does not require who knows what kind of accompanying music.

It is inexplicable how this game is almost unknown outside Japan and how little success it has achieved. We are faced with a pearl for this console that I recommend everyone to try.

On the network you can also find a patched version in English, is not decisive but if you want to understand the dialogues can be useful.

by **Roberto Del Mar Pirazzini**

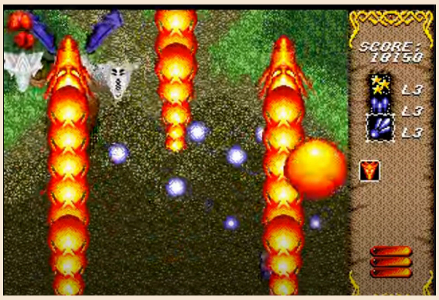
OUR FINAL SCORE

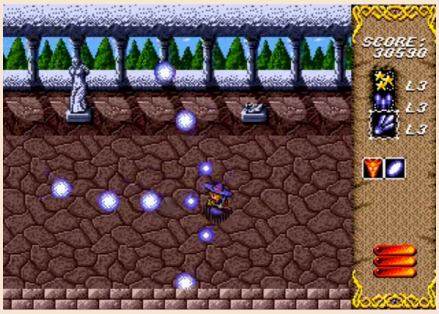
» Gameplay 90%

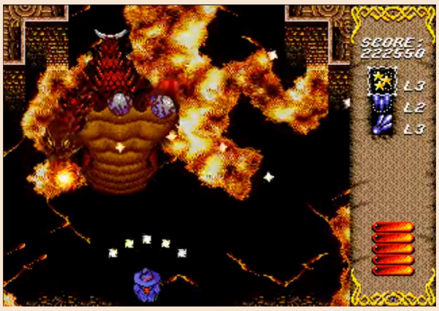
We are facing a gem for this console that I recommend everyone to try.

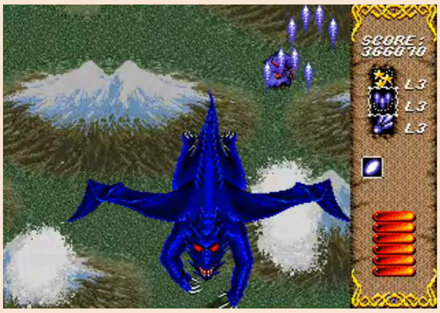
» Longevity 80%

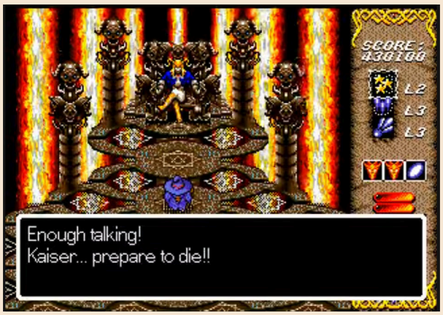
Really stands out in Twinkle Tale the level of variety in its











Enough talking!
Kaiser... prepare to die!!

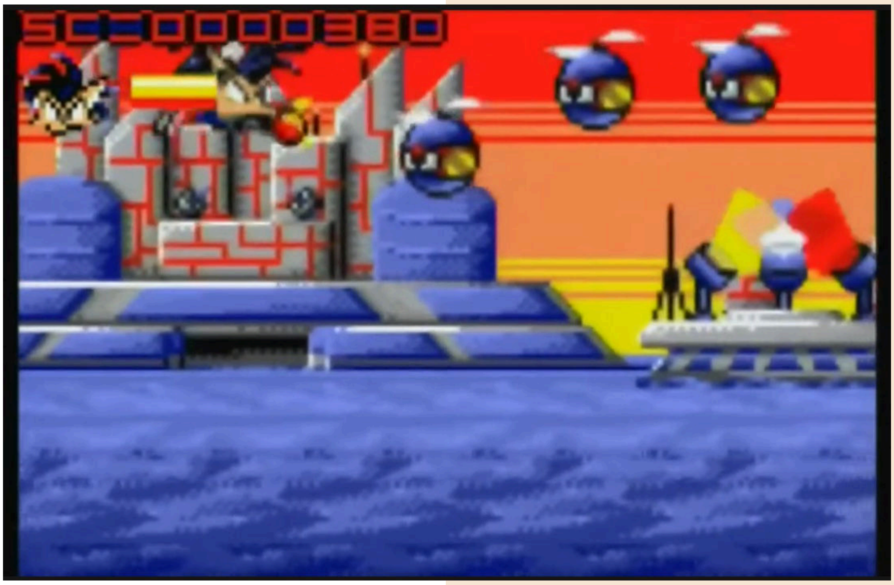
ZAKU

Year: 2009

Editor/Developer: Super Fighet Team - PenguiNet

Genre: Shoot'em up

Platform: Atari Lynx

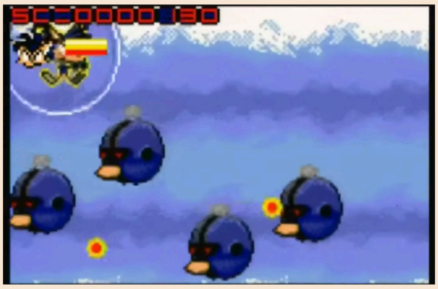


Zaku is one of those games that if it had been released in the golden age of this console would have jumped to the top of the charts of appreciation and sales, perhaps relaunching the portable in the market dominated by rivals Nintendo and Sega.

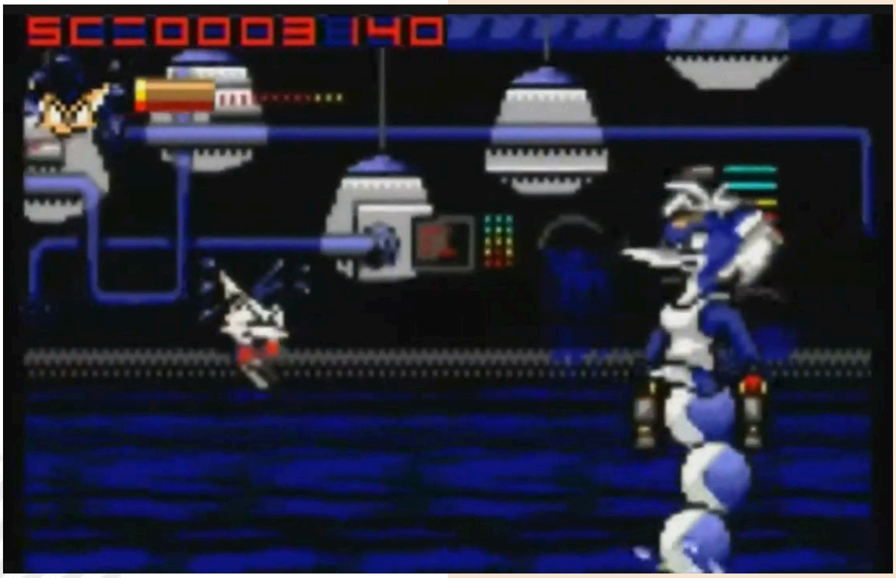
But Zaku is a new game, released only in 2009, about a decade and a half after the Lynx left the market. Developed by PenguiNet's Osman Celimli and published by Super Fighter Team, it's a horizontally scrolling shooter reminiscent of the beautiful Air Zonk for PC Engine.

Despite its homebrew nature, it had a really important commercial launch, starting right from a real development kit specifically for the Atari portable. They put it up for sale in a cartridge and artfully packaged it with a well-made instruction manual.

The game is a real homage to the early shooters of the nineties. The character is characterized just like the various mascots of that period and is reminiscent in appearance of the irreverent style of Sonic and Bonk. The graphics are colorful and well done and there is a parallax scrolling really excellent. Really admirable work. What I love about this game is its huge amount of boss battles. I love fighting these huge, absurd characters. There are at least three bosses per stage, with the final level consisting entirely of multiple consecutive boss fights. Sure some of them are similar, but there's still so much variety that they always feel original.



Some of my favorite villains are, for example, the blue collar fish, the flying



toaster that shoots croutons, and a computer program that continually mass-produces other mini bosses. Brilliant!!!

I also find that it comes with a crazy but enjoyable storyline. It's a fun story about an evil game developer who has gotten his hands on prototype software and is using it to flood the market with games of questionable quality. So it's up to us as Zaku to recover the software and all its copies. An absurd plot from the final really well done and fun that is worth seeing.

Zaku is a clean game, perhaps too clean. There are no real power ups, nor is there too much action on screen. Enemies often fly towards us no more than two at a time, and projectiles aren't as frequent as in other titles in this genre. While this suits the Lynx's small, blurry screen, it also means that the game loses some of the frenzy that's typical of this kind of game.

In addition, some levels are absurdly simple while some bosses will make you list the major planetary gods so difficult.

The game is finished in about two hours, it may seem short but I remind you that the battery life of the console is around three to six hours of activity and that there is no rescue system or password. I do not think it's worth extending the overall duration of the title. Anyway, the longevity is assured by the degree of challenge. Don't worry. Homebrew games like Zaku are a godsend for this little console. It's a special title and probably the best shooter ever played on Lynx.

It's worth seeking out and playing.

by **Roberto Del Mar Pirazzini**

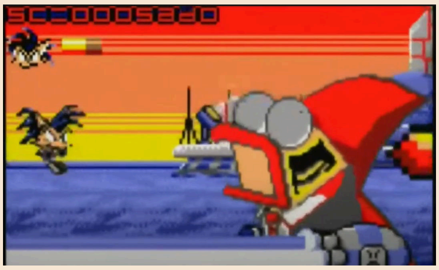
OUR FINAL SCORE

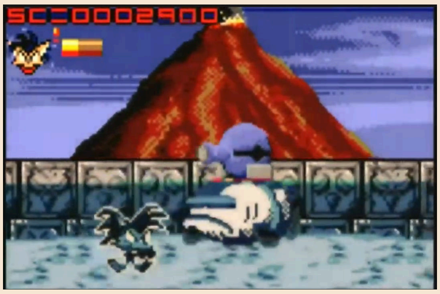
» Gameplay 90%

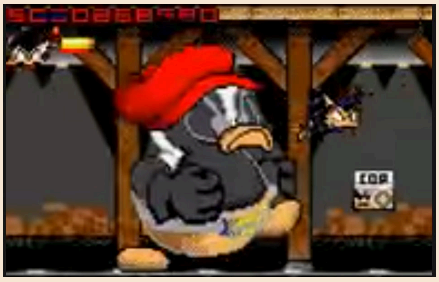
Perfect controls, great gameplay and beautiful boss battles!

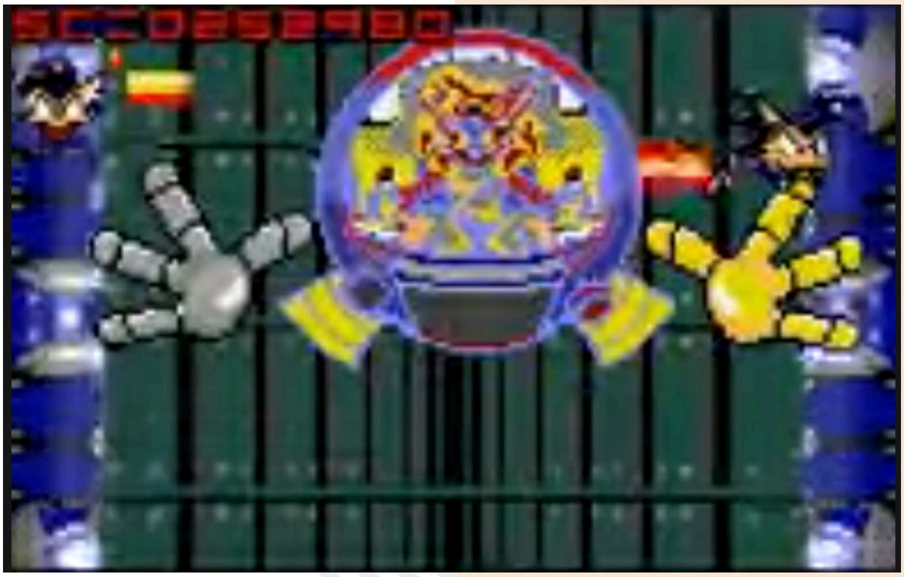
» Longevity 85%

Maybe short compared to expectations but the difficulty is balanced between super easy levels and really challenging boss battles.





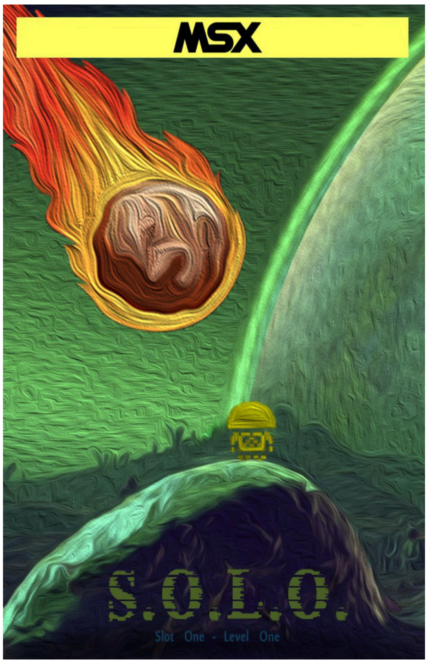




NEW GAME

S.O.L.O.

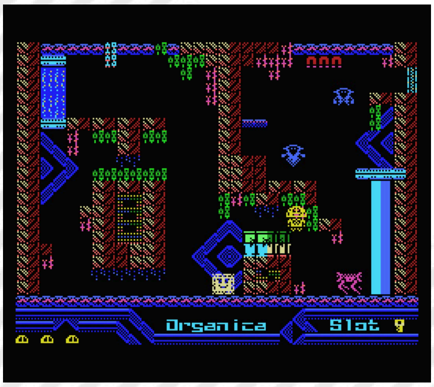
Year: 2021
Editor: Rolandoo
Genre: Platform/Puzzle
Platform: MSX
Website: https://www.msxdev.org/2021/07/16/msxdev21-19-solo/



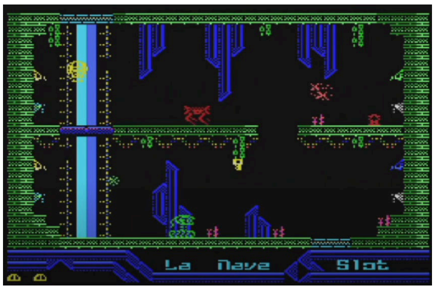
A huge asteroid is on a collision course with the planet. The droids called S.O.L.O (slot one, level one), created for the extraction of a precious mineral called Ether, are the only ones who can prevent the catastrophe.

To do so, one of these robots, the protagonist, will have to make his way through numerous levels full of puzzles and dangers and save himself and the entire world.

We are faced with an excellent title born for the MSXdev last year and that deserves to be played.

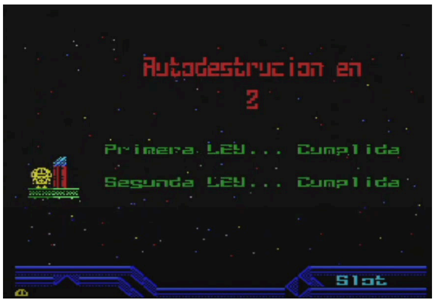


It has an atmosphere typical of this genre of platformers with puzzle tips. The levels are well structured and require a good familiarity and dexterity. Our robot needs upgrades in order to reach the exit of each level.



Simple to play and never punitive, S.O.L.O. is also a little graphic gem with a good audio part.

There is also a version for ZX Spectrum also in this case well done.



Autodestrucion en 2

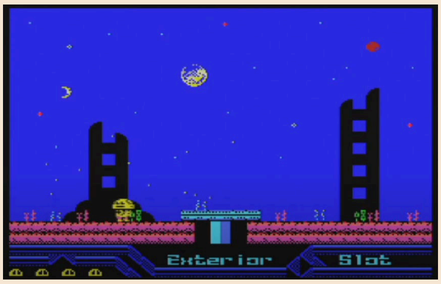
Primera LEY... Cumplida

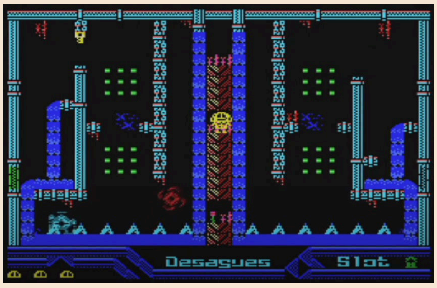
Segunda LEY... Cumplida

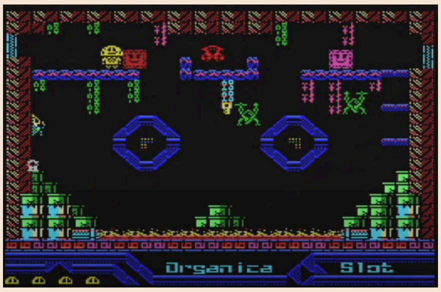
I have to admit that I was "hooked" on this game for several hours so as to complete it with satisfaction. It is not a simple game but it glues you to the joystick.

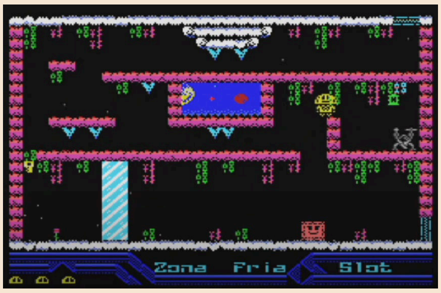
Highly recommended.

by **Giampaolo Moraschi**









OUR FINAL SCORE

» Gameplay 90%

Simple and straightforward. The levels are all well crafted and never nonsensical.

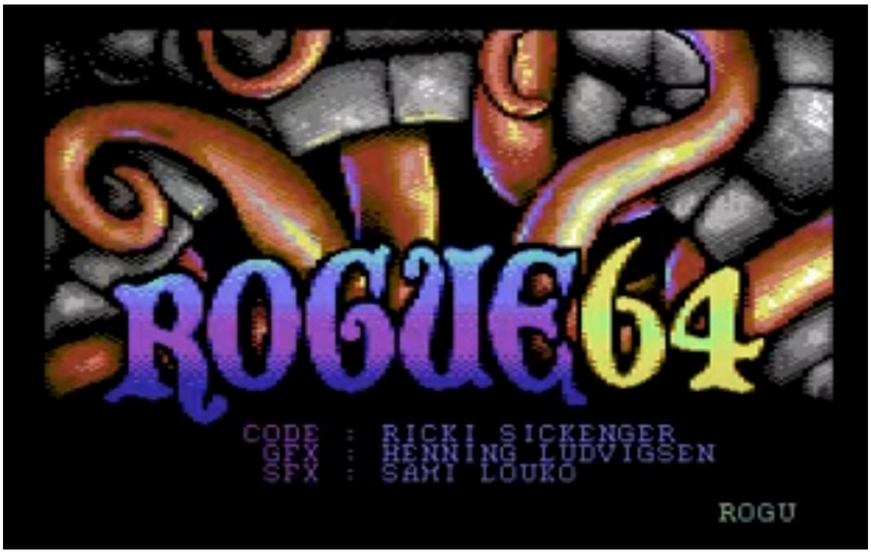
» Longevity 80%

Difficult if you underestimate it. But it will keep you glued to the table to see the end.

NEW GAME

ROGUE 64

Year: 2022
Editor: Badgerpunch Games
Genre: Rougelike games
Platform: Commodore 64
Website: https://badgerpunch.itch.io/rogue64



Available from Bitmap Soft and developed by Badgerpunch, Rogue64 is a modern roguelike adventure that is well worth a try.

The game uses procedural level generation and each game is a unique experience. More complex monsters and more difficult challenges are presented as the game progresses, ensuring that this is a memorable gaming experience on our Commodore 64.

We will be Zendar the explorer, seeking fortune and glory in the dungeon of Mordecoom. It is said that a powerful magical object is hidden in this dungeon and, like all explorers, we long for it!

The only problem are the evil beings that dwell in the caves lurking in the darkness, waiting to attack any hostile visitors.

The game is based on Rogue4k, a similar game that was created for The C64 'Cassette 50' charity contest and took up just under 4000 bytes.

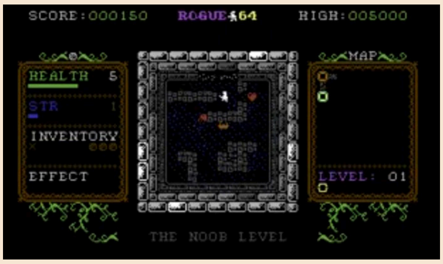
The title has a minimalist but functional graphic appearance accompanied by an excellent soundtrack by Sami Louko. The joystick in port 2 controls the movement of the protagonist; to grab loot just walk on it and to fight enemies just walk towards them.

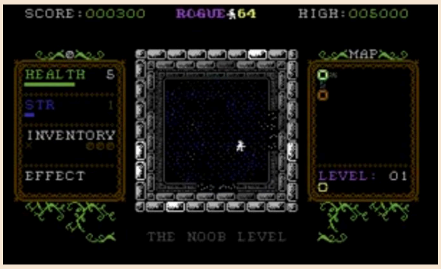
Beware... when you say it like that it sounds like a piece of cake, but I assure you that the level of mortality caused by wrong choices is very high. However, this does not lower the interest curve for the title, which proves to be one of the most valuable of the genre on the C64.

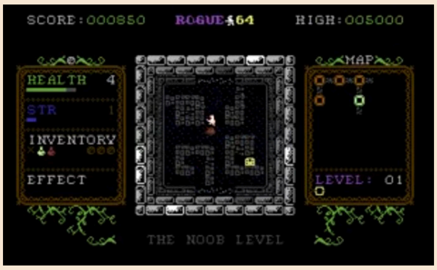
You can buy the title as a digital download on the website or you can buy the Deluxe box for £35 on the Bitmap Soft website. The box includes the cartridge, manual and a range of gadgets.

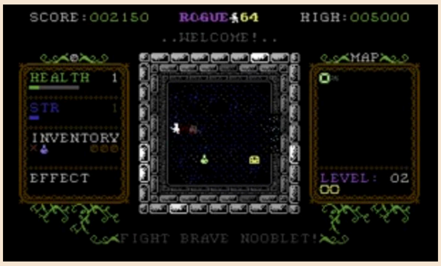
All in all, a good, fun title for the Commodore 64.

by **Carlo Nithaiah Del Mar Pirazzini**









OUR FINAL SCORE

» Gameplay 90%
Essential but addictive control system. Procedurally generated levels increase the desire to play.

» Longevity 80%
Difficult but enjoyable. After the game over you will hit the fire button again and again.

SIGIL

Year: 2019
Editor: Romero Games
Genre: FPS
Platform: PC
Website: https://romero.com/sigil

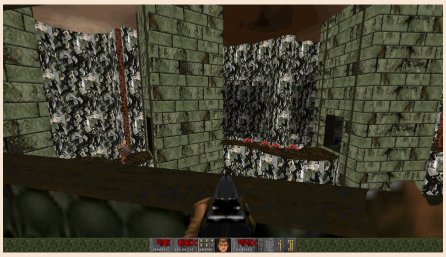
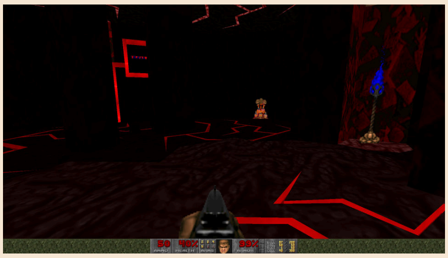
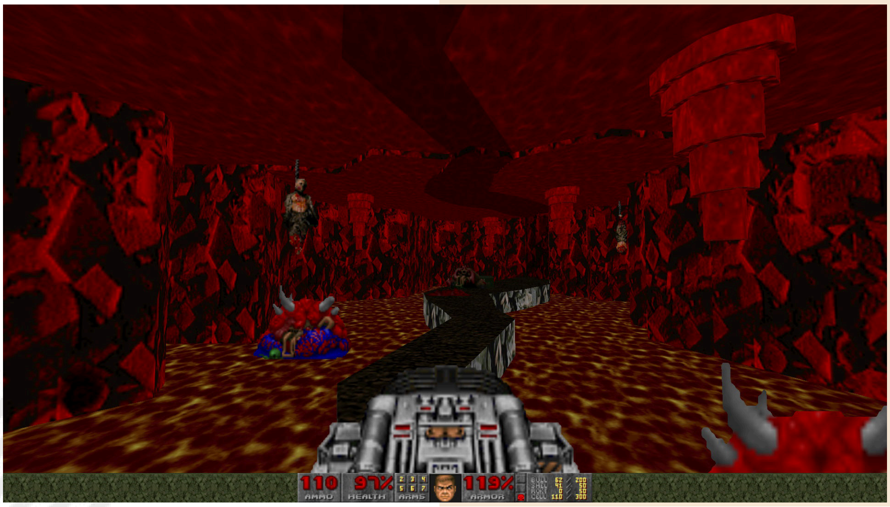
Imagine if one day one of the 12 apostles came to your house and gave you an extra piece of the Bible.

This is essentially what Sigil is to FPS fans. It is the manna from heaven for this genre of games, although given the context, hell might be a more appropriate starting point for this metaphor.

Sigil is an "unofficial" fifth episode of Doom (or Ultimate Doom), though it's as much official as unofficial, created as it is by the one DOOMDAD aka John Romero himself.
This MegaWad adds to the game nine new maps for single player and nine for multiplayer, all created by the fervid imagination of JR.
You can download it for free, but if you want for only 7 pounds you can buy the special version that includes a custom soundtrack.

For all those interested in old-school FPS, Sigil is an interesting product, indeed exciting.
I have to admit to some trepidation when I first heard about Sigil.
Happy with John Romero's involvement, but at the same time terrified. You see just because someone is bringing you a new piece of the Bible, or a new sequel to a TV series or even a new trilogy means that something is good. Take for example "Saint" George Lucas, he gave us new fragments of his "Space Bible" and look how that turned out!
Artists, like all people, often change over time and their creative eye often changes with it, sometimes in a way that puts them at odds with the creator. This is not necessarily a negative trait, but it makes it difficult to recreate the initial atmosphere, the perfection of the first creation... as for example the Star Wars movies or the Alien series have demonstrated.

Add to this that J. Romero was not exactly the most prolific creator in recent years and that Doom needed one more episode like we need another Terminator movie, and Sigil seemed to rest on shaky foundations.
Fortunately, the game is excellent. Not only is it beautiful and satisfying, but extremely fun and with a sequence of challenging, disturbing (sometimes really disturbing!) and very playable maps.
It reminds us of what made Doom exceptional in 1993.
Sigil throws us into the fray from the first few seconds, throwing us into a red room with a pentagram carved into the floor, assaulted by three Imps.

Immediately we are put under pressure, feeling on the brink of death, a theme that continues throughout Sigil.

Sigil is the original form of Doom. Forget survival fps or graphics-pumped sequels. This is Doom in its most hellish and hostile form.

It's a terrifying METAL massacre!!! It's the dystopian-demonic realm birthed by the mind of John Romero, made of guts and blood, walls with faces and entire rooms that writhe.

It's worth noting that Sigil is full of intense firefights. Every room is filled with enemies to blow up. But it doesn't just crash us into entire legions of enemies, it carefully deploys specific combinations of opponents that will make us think.

We'll have to act thinking about the best way to get out of this hellhole.

My main criticism of the game is that it adds no new weapons or enemies. Designed as a celebration of the original Doom and all the parts in it, which is fair enough.

However, it would have been nice to see something new, just to spice things up.

Finally, and this should be obvious, but if you've never played Doom before, you should do so before venturing into Sigil.

It's a game for Doom veterans, with a high degree of challenge that perhaps new players might find "overwhelming".

Overall, I enjoyed the game much more than I expected. Something more than just a tribute to a classic; it mixes and augments all the elements that made the original great, resulting in a rare example of a long-delayed follow-up that makes a significant contribution to the original work.

by **Carlo N. Del Mar Pirazzini**

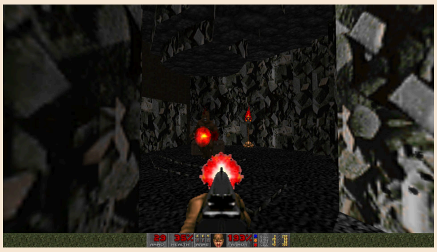
OUR FINAL SCORE

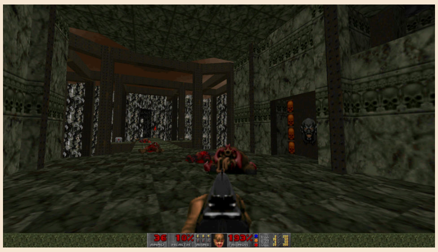
» Gameplay 99%

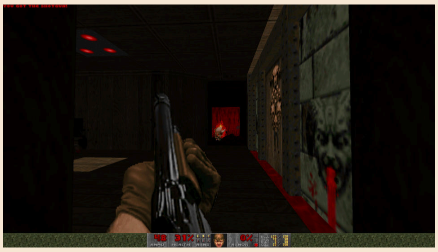
New maps with the same amazing and fast-paced gameplay!!!

» Longevity 90%

Difficult, cruel, hostile and alienating. It's not perfect but it's a masterpiece.







NEW GAME

WITCH N' WIZ

Year: 2021/2022

Editor: Matt Hughson

Genre: Puzzle

Platform: Nintendo NES

Website: bit.ly/witchnwiz

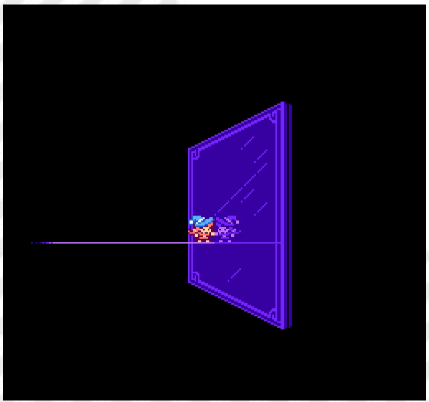
What a fun game this is!

Witch 'N Wiz presents a young witch who is out to rescue her friend, who has been kidnapped by the evil Witch of the Dark.

After the 'playable' introduction, each chapter is divided into several levels.

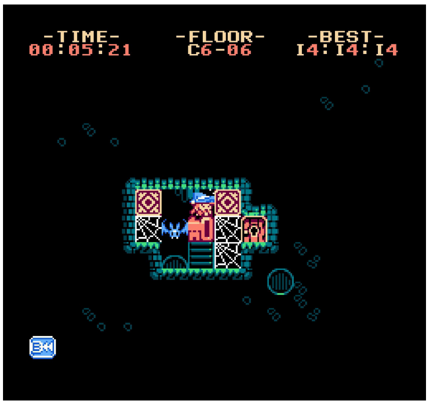
Our chubby witch can climb and push blocks subject to gravity. Their correct positioning determines the success of the level and the passage to the next one!

If you make a mistake, you can rewind the level with the B key and try again.



The degree of challenge of the puzzles for each individual game frame is well structured. Sometimes it doesn't take very long to get to the next stage, other times you'll have to rack your brains to try and solve them, but they're never impossible.

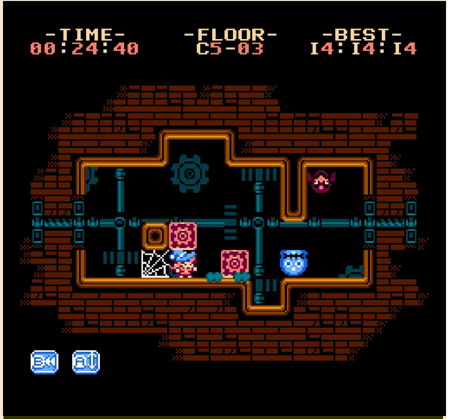
Nice background graphics and very 'KAWAI' characters.

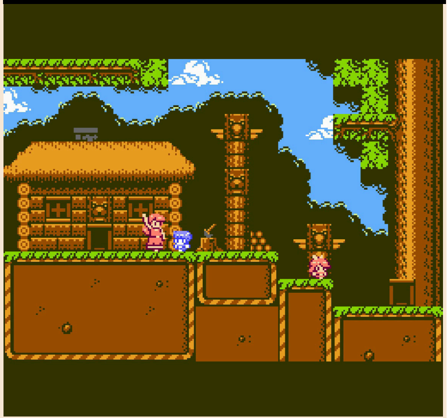


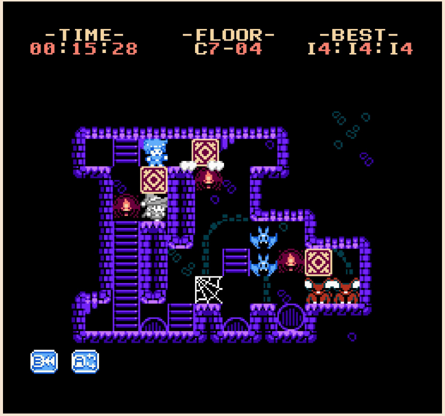
The cartridge version is well edited with a colourful manual and excellent packaging.

I had fun.

by **Ingrid Poggiali**







OUR FINAL SCORE

» Gameplay 87%
Simple and fun to play. No learning difficulties.

» Longevity 87%
A nice degree of challenge! It will keep you entertained for a long time.

DORAEMON

Year: 1998

Editor: Epoch

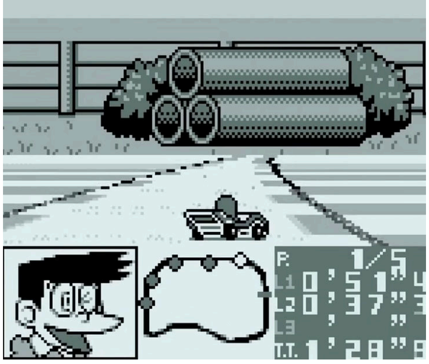
Genre: Driving game

Platform: Game Boy

Ok ok, I have a real problem with Doraemon!

I mean I like the space cat a lot and this is one of those few kart games to have in the collection.

By the way Fujiko F. Fujio's manga is from 1969! Did you know it? Surely the super nerds did (you can be sure dada, NdNith)! Then we all basically watched the anime on TV. But let's get to the game!



Like any Nintendo Kart game, you can take items to use during the race, but do not expect great graphics, it is treated but it is still a small Game Boy!

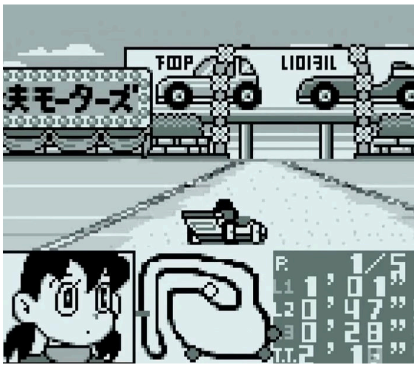
You can choose the various characters: Doraemon, Nobita Nobi, Shizuka Minamoto, Takeshi Goda and Suneo Honekawa, Dorami is unlockable.

You can play multiplayer and that is fun.

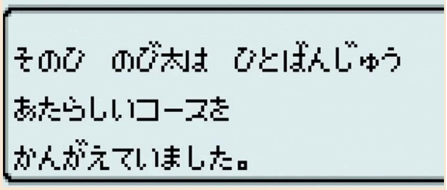
Perhaps it is not very long or difficult, but fun.

On Game Boy Color there is the 2 if you can not stand the greenish / gray tones that offers the GB.

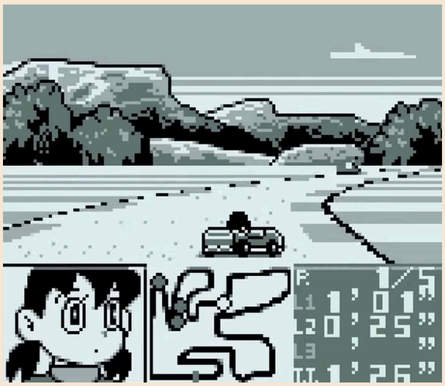
Have you ever played this one?

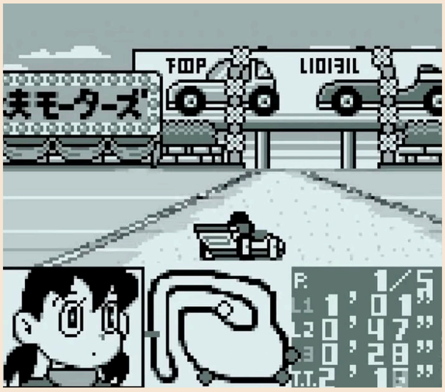


by **Barbara “Morgana” Murgida**



そのひ　のび太は　ひとばんじゅう
あたらしいコースを
かんがえていました。





OUR FINAL SCORE

» Gameplay 80%
Simple and enjoyable to play.

» Longevity 65%
It doesn't take that long to see the final screen of the credits, however, there is multiplayer that extends the life of the game (playable with cable if you have two Gb or with Kaillera in emulation, NdNith).

SKI

Year: 1983

Developer: Commodore

Genre: Sport simulation

Platform: Commodore 64

Spring is just around the corner and there is a feeling of rebirth, it is no coincidence that the Italian national rugby team won the match against Wales at the Six Nations after an avalanche of consecutive defeats; could this be a good sign for this underrated sport in our country?

For being the first day of spring, in Milan it is still cold unless you are in the early afternoon under the Duomo. Thanks to this somewhat anomalous cold, (or maybe the last of the season is the strongest?) I remembered another sport that is little followed except in the days of Alberto Tomba. I'm obviously talking about skiing, with all its categories, such as giant slalom, special, combined...
Exactly like the Ski for Commodore 64, even if, here there is only one category and for a more complete game of this kind you have to wait many years more.
Any sport that comes to mind, you can be sure that on our breadbin something was made at the time, with more or less satisfactory results.

On a global level this game is very entertaining as the music that comes along; six challenging levels in which you must reach the finish line within the allotted time, overcoming obstacles such as trees, rocks, hares that cross the path, without forgetting to pass between the flags! I'm not very experienced in skiing but if you miss them there is a time penalty resulting in overrun and disqualification.

Six levels of entertainment, manual labor and patience that will keep you and your friends busy for entire weekends and holidays.

The first level becomes quite easy after the second game, while the following ones are for real experts of joysticks and timing.
The tracks could lead you to think that it is a repetitive game, but they are rich in details with several small differences. For example, the hare that crosses the path where, in order to save its life, you almost sacrifice your own against a tree... Also, the arrival times become shorter between one descent and another.

Until now, I had only noticed similar games on modern computers or within Olympic games such as Epyx's Winter Games. This makes it special, although it could be better exploited from the competitive point of view.

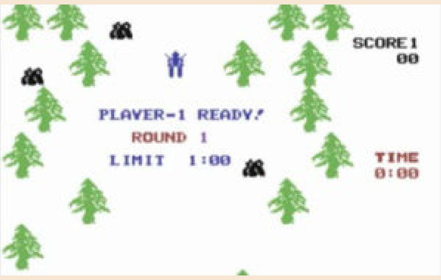
Graphically I do not feel like criticizing it because for a game of the 80s, the graphics were already good. We will have to wait for big productions to see real technical improvements.

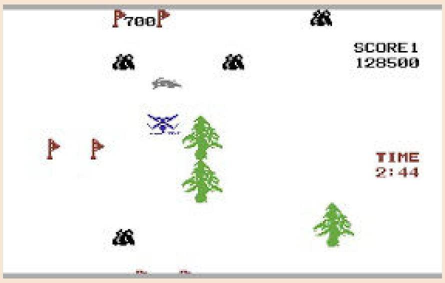
Nice, but not at the highest, the sound that serves more than anything else to keep company in a game like this, especially in case of victory.

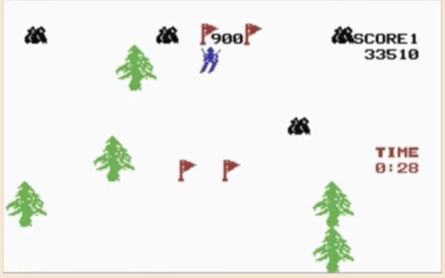
If you try to play it without music and with only the sound effect of the crash, you will notice the difference.

I hope that with this beautiful ski simulator you can calmly say goodbye to winter and above all enjoy the variety of titles that our glorious and worldwide magazine has been offering for years.

by **Daniele Brahimi**







Tribute to Alberto Tomba

OUR FINAL SCORE

» Gameplay 75%
Entertaining... Alone or with friends.

» Longevity 70%
Six tough levels can entice or demotivate. It's up to you...

CRACK DOWN

Year: 1989

Developer: SEGA

Genre: Top Down Shooter

Platform: Arcade

In the vast panorama of arcade games, there is a title that has remained little known but that still maintains a great playability, its name is Crack Down and was published in 1989 by Sega.

The protagonist of the story is Mister X, a mad scientist who decided to create humanoids to conquer the world. That's when Andy and Ben make their appearance, two special agents who must infiltrate various areas before reaching the scientist's laboratory and destroy it.

The game is a top down shooter with the classic top-down view, in perfect Gauntlet style. It presents a real original HUD, with the screen divided into two parts, each for each player, with a lot of indicator of the screw, the various weapons available and a nice map in the center that shows us the various strategic points.

The main purpose of the game is to place different bombs within each level. Triggered the last device, we then have a limited time to reach the exit route before the explosion.

Before reaching Mister X's laboratory, we have to cross four worlds, each consisting of four different areas.

At our disposal we have a rifle, a missile launcher that we can collect along the way and some bombs that destroy everything on the screen. If we finish the bullets, we have to advance by kicking our way forward.

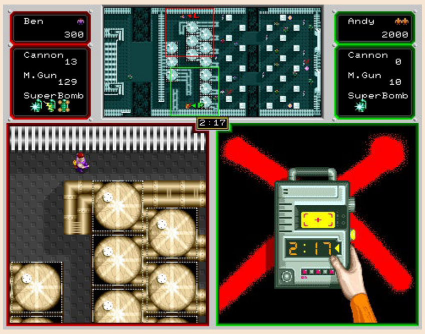
The game is halfway between a shooter and a strategic, in fact we can reason and find alternative routes to place all our bombs. Of course the timer that runs is not our ally and we must be skillful to apply our strategy.

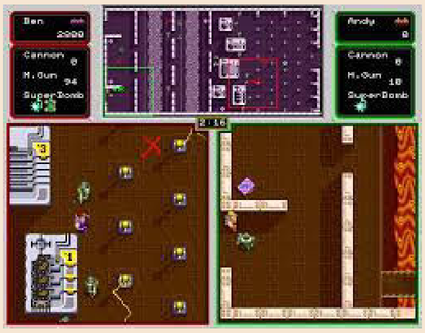
Although starting from an idea already seen, Crack Down has a great playability combined with a pixel art graphics really well done and music and sound effects that blend well with the architecture of the game.

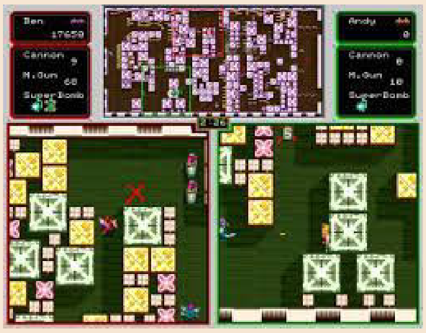
At its release in Japan, the magazines of the period have reviewed it with excellent ratings, even venturing to say that it has an addictive gameplay. In fact, the more you play it, the more you want to continue the adventure to reach the final level. If you play it in cooperative mode, the fun is even more addictive.

Crack Down has received several conversions for consoles and computers. Really noteworthy those for Megadrive and Amiga that, although with different graphic nuances, have managed to maintain the same original and exciting gameplay.

by **Querino Ialongo**







OUR FINAL SCORE

» Gameplay 90%

They wrote that this title has an addictive gameplay. Indeed, its playability is incredible, even more so when played cooperatively.

» Longevity 90%

Like most arcades, Crack Down has a longplay that hovers around thirty minutes, but that doesn't detract from its beauty at all. This title deserves to be rediscovered, despite the fact that some 35 years have passed since its release.

'ARE YOU A REAL RETROGAMER?'

As many of our readers (you are very nice, thank you!), I usually attend several groups on the well-known blue 'F' social network. Groups created and populated by those who, like me, are passionate about video games, more specifically about retrogames, retrocomputers, retro consoles and anything else related to this fantastic world and that begins with "retro". They are often an excellent source of news, insights, virtual places to discuss, where someone suddenly brings up that game you had removed from your memory, where you can find technical information and curiosities. Or simply a place where you can indulge in memories of the good old days, talking about this or that game of the past and telling other people about that incredible record that you set in 1991!

All very nice but it also happens, not infrequently, to run into what I might call the "Integralists of retrogaming". Apparently, they are those who can assign you the license of a true enthusiast - a license that they obviously hold already. They are the ones that tell you how it is really done. They are also the kind of people you can actually find anywhere in real life, not only in the world of retrogaming and, to be honest, you do not meet them only on social networks, but let's say that in those places it's much easier.

Let me give you an example, just the last one I saw: finally the A500 Mini is about to be released (at the time of writing, when you'll read this, it will already be on the market), which follows the trend started by the Mini NES, then continued by the Mini SNES, the Mini C64 and so on. I have read and I still read several criticisms of the product itself, in my opinion some legitimate and some not much so, but I do not want to dwell on this now (we will talk about it in the next issue of RMW, guaranteed!).

What I find really strange is the assignment of the badge of the 'true retrogamer'. From what I read among the huge amount of posts I can deduce that: if I play with one of these mini consoles I am not a true fan, I am even a step below those who use emulators, because I spent money that I could have saved. Then someone comes up and says that no, if I don't play on real hardware I am not a real fan. But then someone else jumps in and says that real hardware is fine, but if I don't use floppies (talking about the Amiga) I am nobody. And no, if the hardware is not exactly what I had 30 years ago, it's not the same thing and probably, with the prices going around today, I've been fleeced as well. The list could go on but I'll stop here, I think you get the idea: there will always be someone "purer" than you, who knows more, apparently equipped with a radar to find the right post to "slip in" and pontificate/polemize.

What often seems to be forgotten is that yes, in short, they are simply games. So let's play instead of talking!

Let's play on PCs, consoles, original hardware but also emulators, smartphones, whatever you want, it's all good. You can use your microwave too if you like and yes, someone really did it.

Let's enjoy our passion without thinking too much about it, let's get invested by that healthy bang of nostalgia that a handful of pixels can give us. Let's go back for a few hours to the carefree kids of the past, maybe challenging our friends in endless two-players challenges. Let's give our sons and daughters the joy that the games "of the past" have given us and let's share this fun with them.

Let's avoid useless controversies, quarrels, endless discussions that lead to nothing. But if we should end up in the middle of all this, and sooner or later it happens, then there is only one thing to do to resolve the issue once and for all and decide who's the winner: a Sensible Soccer challenge? In short, dear readers, let's have fun, let's laugh, let's hug each other and love each other a lot. Let's play as much as we can without worrying about anything else, after all, that's why we do it, isn't it?

Our beloved video games offer us the opportunity to escape from everything at least for a while, put aside for a moment the ballast made up of a thousand problems that all, more or less, we carry around every day, making us immerse ourselves in colorful and often absurd worlds, where the only thing that counts is to overcome the next level.

All this enclosed in a handful of bytes, not bad at all, right?

Beppe Rinella

Disclaimer

RetroMagazine World as an aperiodic magazine entirely ad-free is a non-profit project and falls off any commercial circuit. All the published material is produced by the respective authors and published thanks to their authorization.

RetroMagazine World - EN
Year 3 - ISSUE 14 - APRIL 2022

Chief Editor
Francesco Fiorentini
Managing Editor
David La Monaca
Editing Manager
Marco Pistorio
Web Manager
Giorgio Balestrieri