PROGRAMMING - HISTORY - INTERVIEWS - REVIEWS - INSIGHTS

Future days are back

RETROMAGAZINE WORLD

ISSUE 23 - OCTOBER 2024 - WWW.RETROMAGAZINE.NET

"RENOVATIO" (Latin for "renovation")

Retrogaming and retrocomputing in general have never been more alive. Some may even doubt this sentence, even though those who usually read our magazine are probably enthusiasts like us; still they may, nevertheless, disagree with me.

Indeed, I might add that the landscape is constantly renewing itself. New hardware and software projects keep springing up and keeping our passions burning, and sometimes lighting them up again.

This is the case of the infamous Commodore 64. Any examples of modern hw/sw designs? Let's start with Commodore 64 SD card readers (SD2IEC), which allow us to use modern SD cards as disk drives for our faster and, more importantly, more powerful Commodore computers.

It is hard to imagine, in the late 1980s, that forty years later we would have such small media in our hands, to be able to store programs and data contained in all the floppy disks we possessed then. And, of course, there would still be space left over. Shall we talk about user manuals or the C64 Programmer's Reference Guide being rewritten, re-paginated and made available in convenient pdfs?

I am not talking about the original manuals but the corrected and revised manuals! For those who are curious here is the reference link: *https://pickledlight.blogspot.com/p/commodore-64-guides.html

Or would you prefer me to tell you about new games that are still being made today, by passionate programmers, graphic designers and musicians, readily available for free or for very little cost? Again, a reference link here, but the landscape is truly vast and ever-changing. https://itch.io/games/newest/tag-commodore-64

Do you like demos, intros, trainers and new software creations in general? .sid tracks, graphic images made for your beloved Commodore and much more? Here is the site, among many to tell the truth, where you can always tap into new content. And there's even something made by yours truly! https://csdb.dk/

Shall we talk about new hardware? I would start with Retrogames' "TheC64," which has elicited so much entuasms and, at the same time, so much criticism.

The excitement stems from the fact, for example, that for the first time since Commodore's disappearance from the market, new devices similar to our Commodore 64 are being manufactured and distributed.

Criticism stems mainly from the fact that these devices, at their core, emulate the Commodore 64 via software and, therefore, the user experience may not be perfect. Here's where to learn more: https://retrogames.biz/products/thec64/ *Going a step further, someone well thought of making a Commodore 64 by exploiting current FPGA technologies, and in this wake, "The Ultimate 64" was born from this idea. The reference link is as follows:*https://ultimate64.com/

But why only accurately simulate the Commodore 64? And indeed, taking advantage of this technology, here is another product, known as "Mister." Link here: https://ultimatemister.com/

Our readers are well aware that we often deal in our magazine with these "pearls" that keep popping up. The point of this editorial of mine is just that. All these insights I have provided serve to demonstrate to you, if there is still a need, how retrogaming and retrocomputing are always alive and evolving so...keep following us, we will have plenty more to see, along with you, my reader friends. Hello and good reading to all!

Marco Pistorio

TABLE OF CONTENTS

The following contributed to this issue of **RetroMagazine World** (in no particular order)::

- Drave Nardella
- Carlo N. Del Mar Pirazzini
- Daniele Brahimi
- Francesco Fiorentini
- Leonardo Miliani
- Roberto Del Mar Pirazzini
- Giampaolo Moraschi
- Giuseppe Rinella
- Filippo Santellocco
- Michele Ugolini
- Eugenio Rapella
- Takahiro Yoshioka
- Marta Rossmann
- Mic the Biker Novarina
- Gianluca Girelli
- Marco Pistorio
- Cover image: **Giuseppe Mangini**
- Cover layout: **Carlo N. Del Mar Pirazzini**
- Closing image: **Fabio de Renzis**

FPGAs for absolute beginners: starting from scratch - part one

by Dave Nardella

Introduction

Let us state up front that this article is not a course in FPGA design. The goal is to identify an educational path for approaching these systems.

The problem with FPGAs is not their inherent complexity; on the contrary, as you will observe: they are too simple. This generates the need to acquire a great deal of information to understand how to exploit this flexibility; and that is the crux of the problem. This information often wrongly considered "all essential right away" goes to overload the learning curve causing many people to desist. In this context, a Step 0 is therefore necessary, purely introductory but with small explanatory insights, in order to be able to identify a direction, but more importantly to estimate how far you need to go. Therefore, I will describe their architecture, programming philosophy, give small practical examples to visualize what is described, recommend the training materials to use, and the first hardware to purchase.

The length of this article may seem to contradict what I have just said, but unfortunately the topic is multifaceted, stopping at simple definitions would only give a superficial idea of FPGAs, and in the end you would know as much as before, I assure you zero.

Unfortunately, there is a layer of rock to overcome in one step after which it is really possible to understand whether these systems can have a future in our interests and what commitment we need to address them.

Rest assured, I use the pickaxe, you just have to follow me.

Summary and prerequisites

To program FPGAs you need to have a clear idea, at least at the system level, of what a CPU is and be familiar with the concept of a development system, meaning software and boards (electronic boards that house the CPUs). If terms such as Raspberry, Arduino, or embedded systems do not ring a bell at all, then it is best to bridge this gap before continuing.

Digital electronics skills help tremendously but are not essential; a Python programmer is not required to know the registers of the CPU that will execute his code; as you will see, there are various levels of abstraction. Programming basics, on the other hand, are necessary.

The macro-topics we will address are:

- Architecture of an FPGA
- Programming and simulation philosophies
- Fields of application
- Brands and families
- Recommended approach
- Purchasing tips
- Recommended resources

Architecture of an FPGA

FPGA is a programmable chip. Its name stands for Field Programmable Gate Array, which is a user-programmable logic gate array.

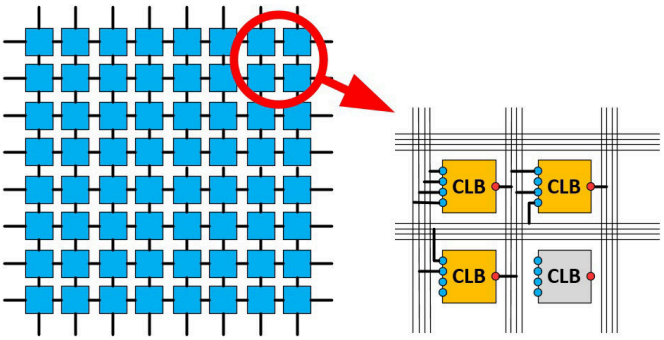
Think of a large tapestry; it is composed of a set of horizontal threads, called the weft, and a vertical thread, called the warp.

The concept behind it is very simple: by using knots of different colors, at the intersection of warp and weft, it is possible to make mind-blowing patterns.

Tapestry is a complex system made by the composition of small elements, all very similar, through simple rules. The FPGA works in the same way, not surprisingly the basic structure takes the technical name "fabric," which means fabric in English. The nodes are represented by CLBs (Configurable Logic Block, which we will see in a moment), small programmable logic units, all the same,

which can be interconnected to horizontal and vertical wires. Programming an FPGA boils down to parameterizing individual CLBs and determining which connections should be made (fortunately, we will not have to deal with this level of detail).

For this reason, it is also often referred to as designing an FPGA; in this context, the terms are equivalent.

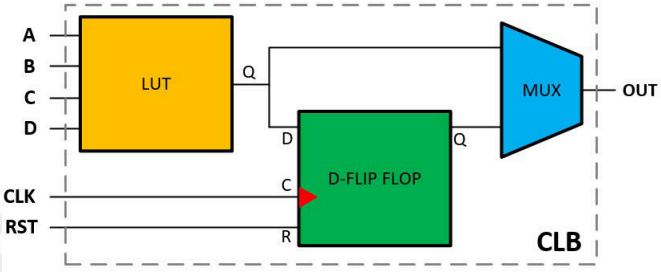


In the figure you see, on the left, the conceptual diagram, and on the right, in detail, the connections that can be made between the various CLBs. Obviously, only the CLBs that are needed will be used.

CLB

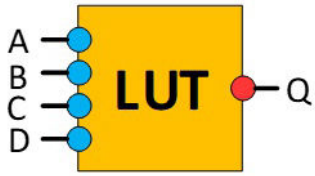
In order to program an FPGA, as we shall see, it is not essential to know their internal composition, but it is only fair to mention it for completeness of information. If you have no notion of digital electronics you can skip this step.

In the simplest form, CLBs consist of a LUT (Lookup Table) that can be 4- or 6-input, a D-type Flip Flop and a Multiplexer connected as follows.



LUTs are small logic units that output a 1 or 0 signal depending on the combination of inputs. Thus, there will exist an association table similar to this one.

D	C	B	A	Q
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
:	:	:	:	:
1	1	1	1	0



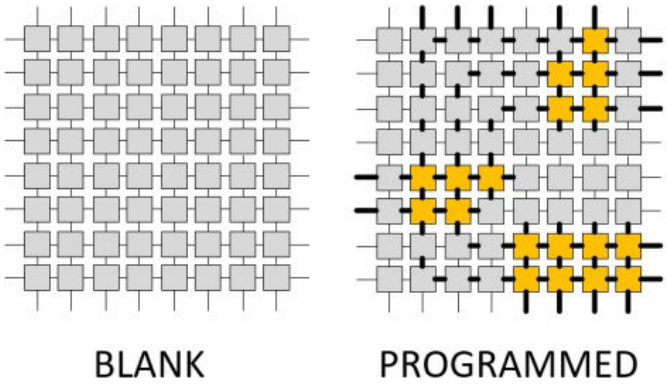
The Flip Flop allows the information at its input D to be stored on the rising edge of the clock pulse, so we can consider it a 1-bit mini-memory.

The Multiplexer is simply a diverter, it outputs the signal present on one or the other input depending on how it is parameterized. In our case we can have at the output either the direct signal coming from the LUT or the deferred (stored) signal coming from the Flip Flop.

Parameterizing a CLB, then, simply means defining the internal table of the LUT and determining what the Multiplexer should output.

Please note, the internal parameterization of a CLB is not done manually; the development system takes care of that. Even when we program a CPU in Assemby, we do not define how to move the Carry bit or fill the Cache.

When we purchase an FPGA then, it will be completely empty.



At this point, remembering what we said earlier, the FPGA is a complex system made by interconnecting and parameterizing small elements, all the same, through simple rules.

Input/Output

All electronic chips have terminals, even FPGAs; in our case these are represented by many (but not all) wires coming out of the four sides of the fabric.

Actually, they do not go out directly; in fact, there are specialized blocks (called IOBs) on the perimeter that allow, once parameterized, to adapt the electrical level, to determine whether the terminals are Input, Output or InOut, and what kind of electrical configuration they should exhibit (push-pull, open drain, etc.).

This parameterization will also be done in the development tool and is specific to the chip we are going to use because the terminals will have to follow the manufacturer's numbering.

Anyone who has programmed Arduino will have used the

pinMode() function; the concept is identical.

Special blocks

What we have just seen is a basic, almost theoretical FPGA; as time has gone on, these chips have evolved.

While using FPGAs, we find that there are circuits of very recurring use; the great advantage of FPGAs is that, starting from basic blocks, we can make virtually anything; we can certainly create libraries with basic circuits, however, it is not necessarily the case that everything we are going to make is convenient (and efficient) to do it that way. It would be convenient to have these blocks already available; this is exactly what happens in modern FPGAs.

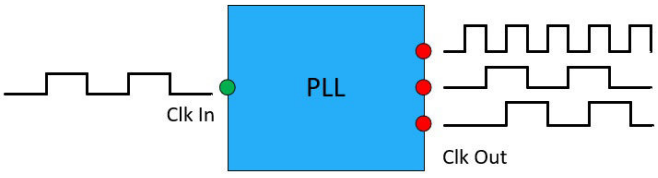
Over time, the presence of these modules has grown exponentially; what might seem like a violation of the basic principle is actually a great convenience, because these modules are also "untethered" within the FPGA and we can decide whether and how to use them; in some cases, even, the development tools are smart enough to figure out when it is appropriate to use them, completely transparently.

The most widely used are undoubtedly memories, DSP blocks and PLLs.

Memories are RAM banks of various sizes and word widths that development tools can easily combine with each other.

DSPs are arithmetic blocks that allow multiplications and sums of integers in 2's complement and are equipped with an accumulator register, usually starting at 18 bits. They are extremely efficient and today it is not uncommon to find them even in low-cost FPGAs.

Continuing, realizing sequential logic inevitably involves the use of a system clock; in reality, very often, we need multiple clock signals, at different phases and frequencies. Here again, to avoid their realization by CLBs (which would also introduce undesirable delays on the clock lines) there are PLL modules which, easily parameterized, starting from a base signal, allow the generation of multiples and submultiples of the input frequency.

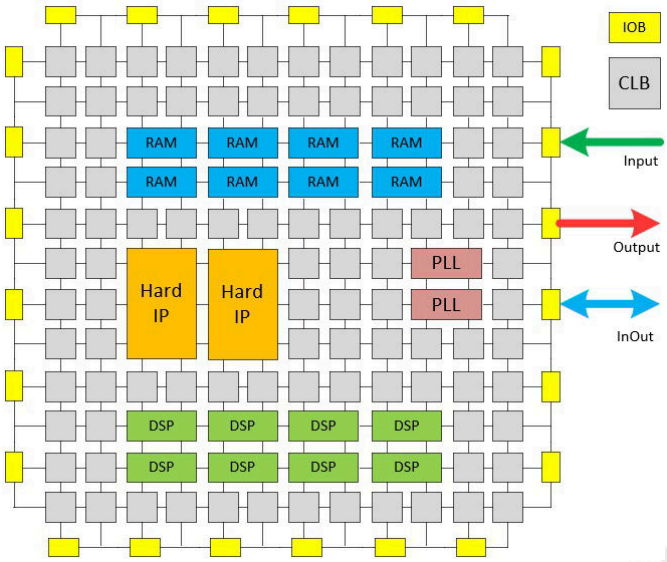


Then there are more specific modules found in mid- to high-end FPGAs, e.g., serdes (Serializer-Deserializer) that allow parallel/serial conversion and vice versa at speeds of several Gb/s. They are used in telecommunications equipment, military and wherever high data transfer rates are needed. But it is also common to find controllers for external DRAM, Ethernet modules, A/D and D/A converters, etc. Anything that can make our lives easier.

Generally, except for memories and DSPs, all extra blocks made in the traditional way, that is, designed directly in silicon, are named Hard-IP (Intellectual Property) because they are implemented in a proprietary way by the individual manufacturers, will be instantiated and parameterized directly in our programs. There are also Soft-IPs, but we will discuss these later.

You may have already realized that the size, and consequently the cost, of an FPGA depends on the number of CLBs and the presence and number of specialized blocks.

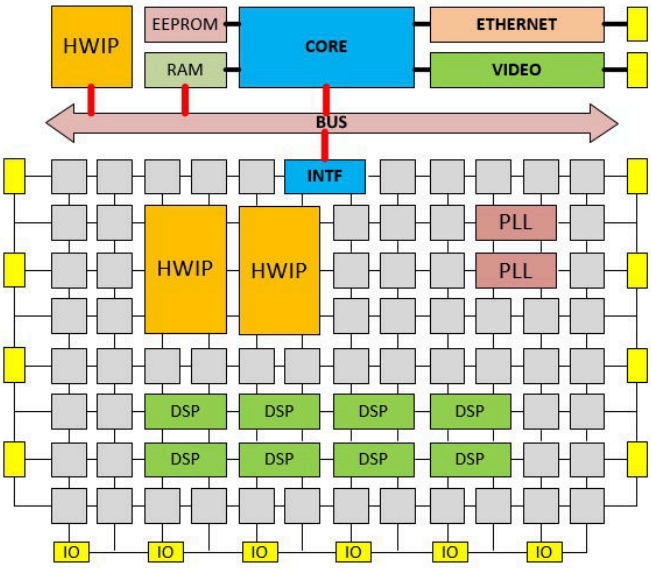
This is a more realistic FPGA in terms of structure, but not in terms of the number of cells, which can go up to hundreds of thousands and even more.



SoC

Finally, as a last evolution, we find SoCs (System on Chip) which are chips containing one or more cores, usually ARM or RISC-V, which are called HPS (Hard Processor System), and an FPGA fabric, all connected by high-speed buses.

In this case, it is difficult to determine whether these chips are to be considered CPUs with an FPGA area to realize complex logic, or super-FPGAs that contain one or more service cores to interface with the outside world more easily.



It all depends on the context of course.

For our purposes we will focus only on FPGAs and the most common extra blocks. The use of SoCs and complex IPs should belong to a second learning step.

Let us summarize what we have seen so far:

- The FPGA is an array of configurable blocks that can be connected together.
- There are special blocks, also to be linked, that efficiently integrate standard and proven functions.
- The "clusters" created with the connections of the individual blocks, being distinct circuits, all work in parallel.
- Programming or (its synonym) designing an FPGA means establishing the parameterization of individual blocks and their interconnections

Let us now see how to design our tapestry.

Programming philosophies

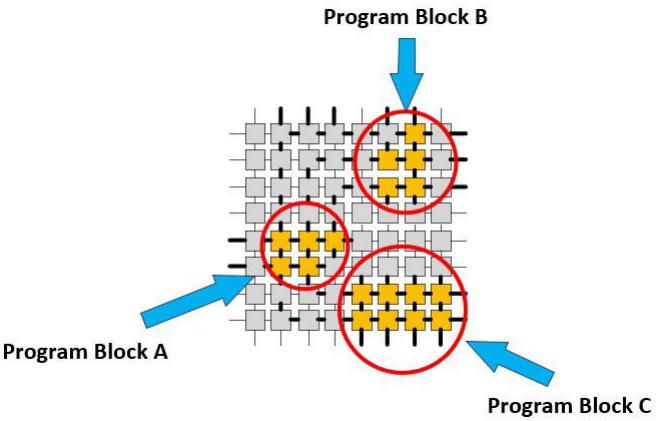
Parallelism

This is a concept derived from what we have seen, and I want to emphasize it right away because it is basically one of the main reasons why an FPGA is often preferred. Groups of CLBs, connected to each other, can have interconnections with other groups or be completely independent, so all these logical clusters, deputed to a specific purpose, will work completely in parallel. Unlike a CPU/MCU where micro-tasks are never completely parallel and must be executed in time-sharing, each task we are going to accomplish in an FPGA by means of a specific circuit will work completely parallel to the others.

Program blocks

Even before going into details, we realized how FPGA programming must follow "special" methodologies.

From what we have seen so far, we could imagine something similar.



With small deviations, that is exactly what happens.

This is quite comforting; as with traditional programming, we can break down our problem into many smaller parts, reducing its complexity.

Another consideration we can make is that our programs will not be a sequence of instructions for a CPU to execute, but must allow us to establish parameters and connections; therefore, we expect one or more languages geared to this purpose.

This is also correct, in fact the languages for designing an FPGA are called HDL (Hardware Descriptor Language), we will use a set of instructions that describe what we want to make, there will then exist a compiler that will translate everything into how it is to be made.

I said at the beginning of the article that to program FPGAs it is not essential to have in-depth knowledge of digital electronics, but so far we have only talked about logic blocks and connections. How do you reconcile the two?

The short answer is that, similar to traditional programming, there are layers of abstraction that allow programs to be made largely independent of the underlying hardware.

The time has come to introduce the four programming methodologies, but first let's introduce the last concept.

Soft-IP

Similar to what we have seen with Hard-IPs, Soft-IPs, or more briefly IPs, are basically library modules. Blocks written in one of the languages we are going to look at, which we will instantiate in our programs as classes and connect them together.

Programming methodologies

Starting at the highest level of abstraction, we find:

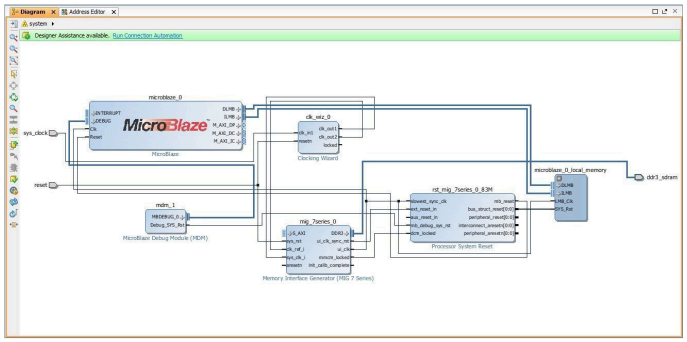
1. System
2. Behavioral
3. RTL (Register Transfer Level).
4. Physical

System

System programming formally is not a separate methodology, as we shall see, conceptually it falls under Behavioral, however, it is convenient to distinguish it especially when we work with SoCs, where HPS and FPGA part are two separate entities.

Basically it is the highest level one where we just instantiate, parameterize and link our IPs.

More advanced tools allow us to do this in graphical form by creating block diagrams, and in this way, we can program an FPGA without writing a line of code.



Shown is a minimal system containing Xilinx's MicroBlaze; MicroBlaze is a softcore, i.e., the core of a CPU made via software, which can run direct programs (called bare-metal) or, having sufficient resources, allows hosting Linux or an RT-OS (Realtime-OS). Many free softcores exist, especially RISC-V, as we shall see.

With this system we can allocate the necessary memory and peripherals; having a powerful FPGA available, we can also think about putting in more than one softcore and have them communicate via an AXI bus (also in the library) like the one used by ARM.

Using a graphical tool, the process is very simple: you enter all the IPs through a trivial insert, link them together, parameterize them through guided menus, and finally you can compile everything.

Where there is no graphical editor, we will use a textual language by going to instantiate our IPs, which we will then parameterize and link via code.

I want to point out that even when programming graphically, the tool will turn the diagrams into text files before compilation.

Behavioral

This level describes the "behavior" of our circuit, that is, what functionality it should have, without going into detail about flip-flops, memories or logic blocks. It will be up to the compiler to recognize the basic blocks to be used, parameterize them and interconnect them.

A Behavioral program will be portable as long as the "external" IPs are also present in the target system and have the same interface.

RTL

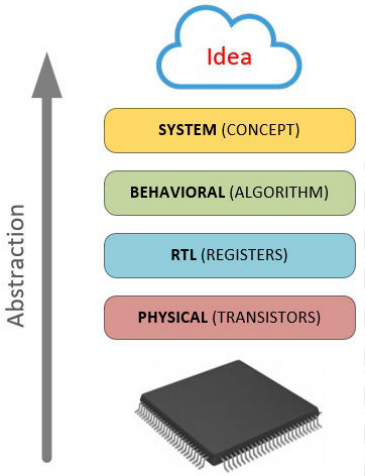
It stands for Register Transfer Level, and as the name states, our programming will be expressed in terms of registers, transfers between registers, and combinatorial logic.

Here we are closer to the hardware, however, still at a higher level than CLBs. What we write will be portable, because the hardware we are going to use will be generic and present in all FPGAs regardless of size and Brand.

To make an analogy, when we work with Files (open, close, read/write), we are reasonably sure that these entities will be present in all operating systems.

Physical

Here we work directly with the hardware of the FPGA we are using, we can establish the location of clusters, indicate preferred paths for clock lines and make all the optimizations, specific to that chip that the compiler is unable to do.



A very thorough knowledge of all timing issues is required as well as considerable electronic training. Don't think about it just yet, remember that development tools are always better able to optimize FPGA code; physical intervention, if not done knowledgeably, could be

counterproductive.

Programming languages

Before giving some examples, it is necessary to introduce programming languages.

There are two categories of languages for programming FPGAs, one graphical and one textual.

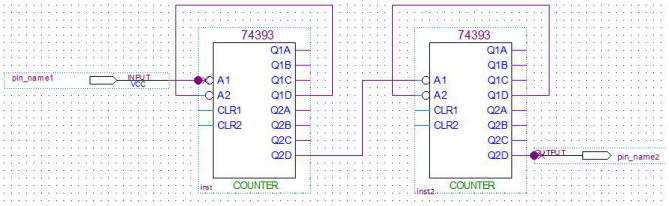
Graphic Programming

Graphical programming does not exist in all development systems and is not standard, that is, it is implemented differently by different vendors.

We have already seen system-level programming; the other level where we can use a graphics editor is the RTL level.

If we want to use an FPGA to make special logic networks, which made in discrete logic would force us to use dozens of different integrated circuits, this methodology is quite advantageous.

We are going to make our circuit diagram by inserting logic gates, flip-flops, counters and whatever else we need. The compiler will turn this design into FPGA code (we will see an example later).



Obviously as complexity increases, this methodology presents enormous management problems, its convenience of use stemming directly from the user's background; for an electronic designer, accustomed to working with logic gates and integrated circuits, it certainly represents the comfort-zone.

One last consideration: everything we accomplish graphically we can express in a standard, portable text language; in fact, all programming tools that implement graphical programming allow it to be translated into text mode.

Textual programming

For textual programming, the two main languages, which we recall to be HDL (Hardware Description Language), are Verilog and VHDL.

They are exactly equivalent; using one or the other is just a matter of taste and leaning toward one or the other syntax. In fact, Verilog is very similar to C, while VHDL is to Pascal/ADA.

The community is split exactly 50-50, each faction trying to convince the other that its language has extra advantages- don't believe it, choose the one that is closest to your programming background.

Verilog is more compact, it has very basic type control, we can copy any variable to any other, even if they have different sizes, just as it is done in C.

VHDL is more "verbose," more words need to be written to express a construct, it is strongly typed, and therefore assignment between different types can only be done by explicit cast (which, however, is only at the compiler parser level and does not produce extra code).

VERILOG

```
module logic_v(input a, b, c, output q);

    assign q = (a & b) |  ~c;

endmodule
```

VHDL

```
library IEEE;
use IEEE.std_logic_1164.all;

entity logic_vhdl is
    port (
        a : in STD_LOGIC;
        b : in STD_LOGIC;
        c : in STD_LOGIC;
        q : out STD_LOGIC
    );
end logic_vhdl;

architecture RTL of logic_vhdl is
begin

    q <= (a and b) or (not c);

end RTL;
```

A big advantage is that all the tools allow you to handle projects containing mixed modules, Verilog and VHDL, so knowing both, we can think about importing/editing any IP regardless of how it was written.

There are also other languages or preprocessors that increase the level of abstraction, such as SystemVerilog or SpinalHDL, or even compilers that allow you to use C/C++ directly, but I recommend that you avoid them at first; start with the basics.

With Verilog and VHDL we can program at any level, even partially at the physical level since there are low-level primitives.

Before giving some examples let us briefly recap what we have seen.

- We can program FPGAs at four levels of abstraction.
- We can use graphic (proprietary) or two highly portable textual languages covering the three most commonly used levels.
- The text languages, functionally are identical, we can choose whichever one we like or even to use both in the same project

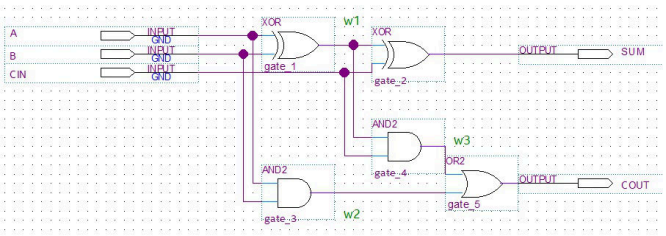
Behavioral/RTL programming example

This is only an introductory/orientative article, I will

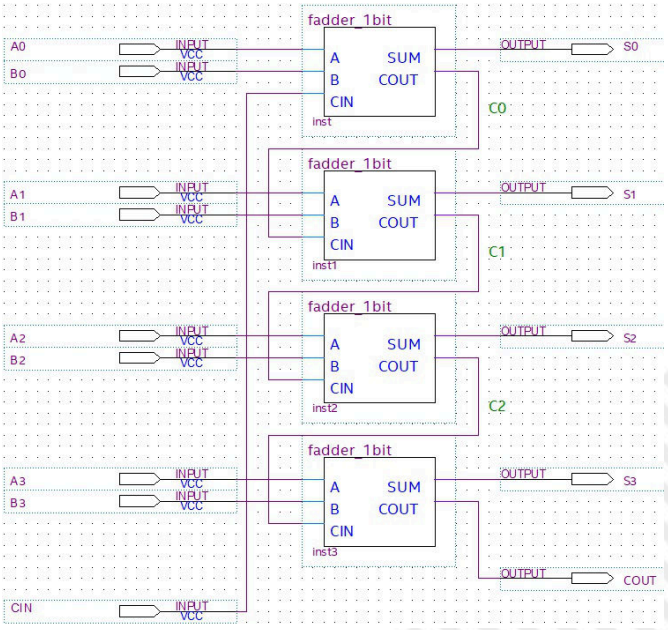
recommend suitable texts and tutorials to learn the two languages; however, let us see an example to understand the difference between Behavioral and RTL programming.
Problem: We want to sum two four-bit words and we stipulate that this must be modular, that is, we must be able to receive as input a Carry In bit (arithmetic carryover) from the previous operation and then output a Carry Out bit useful for the next operation. This will eventually allow us to concatenate the blocks we are going to create in order to add words of greater length.
For the record, in "digital" terminology, a block that adds two words together by handling Carry In and Carry Out is called a full-adder.
We will use both Verilog and the Quartus Prime graphics language (the Intel-Altera development system).
Let's start with the Behavioral approach. To add up two words we will write something like this:

```
module fadder_4bit(
                input[3:0] a,     // 4 bit (from 3 to 0) operand A
                input[3:0] b,     // 4 bit (from 3 to 0) operand B
                input cin,        // Carry In
                output[3:0] sum,  // 4 bit (from 3 to 0) Sum
                output cout       // Carry Out
                );

   assign {cout, sum} = a + b + cin;

endmodule
```

This is a Verilog module, conceptually it is very similar to a class: if it is not the main module of the project, it needs to be instantiated in order to use it, and this, as with classes, we can do it as many times as we want.
The input and output words are arrays of four bits declared in the form [Final bit : Initial bit], all variables, unless specified, are wire type i.e., simple links that do not store information after an assignment, as opposed to reg type variables, in which the value remains even after assignment.
Another peculiarity we encounter when looking at this code is grouping; referring to the construct {Variable A, Variable B} is equivalent to creating in memory a variable that becomes the geometric concatenation of both variables listed; by definition: top - bottom, left to right.
In our case, to the variable sum, we are going to concatenate a bit (the Carry out) that will be filled if the sum a + b + cin produces a carry out, that is, if it is greater than four bits. Ultimately, the concatenation {cout, sum} is a 5-bit wide variable.
Observe that if we did not have the type declaration, which is necessary because we created a module, the whole thing would be reduced to a single statement.
We come to RTL programming. We are going to create a four-bit adder as a concatenation of four one-bit full-adders, this allows us to decrease the total complexity by using a hierarchical approach, as will become clear especially by looking at the graphical diagrams.
We will see how to do this both graphically, with a wiring diagram, and textually.
This is the schematic-file of a one-bit full-adder. (The labels w1, w2, w3 are just explanatory labels that I inserted whose reason will be clear later)



This circuit sums two bits, A and B, considering the input carryover CIN, and outputting, if appropriate, the output carryover COUT.
If we want to extend the word, we need to use multiple adders by cascading CIN and COUT.
Following is the schematic of the four-bit full-adder. To implement it, I created an IP from the one-bit full-adder and inserted (i.e., instantiated) it four times then made the necessary connections.



Let us now look at the textual counterpart in Verilog. I have purposely used the library primitives of the logic gates to show the exact analogy with the wiring diagram, and now the labels w1, w2, w3 which are temporary variables representing connecting wires will make sense.
If we had not wanted to use logic gates, much more simply, instead of and gate_3(w2, a, b); we could have written

assign w2 = a & b; the result is the same.

By definition, the first parameter of a logic port is always the output, followed by the inputs.

```
module fadder_1bit_rtl(
                       input a,
                       input b,
                       input cin,
                       output sum,
                       output cout
                      );
    wire w1, w2, w3; // temporary

    // gate_x is an arbitrary (unique) instance name
    xor gate_1 (w1, a, b);
    xor gate_2 (sum, w1, cin);

    and gate_3 (w2, a, b);
    and gate_4 (w3, w1, cin);

    or gate_5 (cout, w2, w3);

endmodule
```

This is finally the full-adder where the four one-bit full-adders are to be instantiated.

As you can see, the two approaches are quite different.

```
module fadder_4bit_rtl(
                input[3:0] a,     // 4 bit (from 3 to 0) operand A
                input[3:0] b,     // 4 bit (from 3 to 0) operand B
                input cin,        // Carry In
                output[3:0] sum, // 4 bit (from 3 to 0) Sum
                output cout       // Carry Out
                );

    wire c0, c1, c2;

    fadder_1bit_rtl adder_1 (a[0], b[0], cin, sum[0], c0);
    fadder_1bit_rtl adder_2 (a[1], b[1], c0, sum[1], c1);
    fadder_1bit_rtl adder_3 (a[2], b[2], c1, sum[2], c2);
    fadder_1bit_rtl adder_4 (a[3], b[3], c2, sum[3], cout);


endmodule
```

Let's recap

- In the first example (Behavioral) we told the compiler what we wanted to achieve, leaving it to decide how to do it.
- In the second (RTL) we also told him how to get it.
- We have seen that graphical programming is equivalent to drawing an electrical diagram and that this can be expressed textually

The big advantage is that these methodologies can coexist in the same project.

Simulation

Debugging the program of an FPGA is not the simplest task; we do not have the ability to insert breakpoints because there is no real sequential execution, and we also do not have the ability to perform variable watch.

Also, for complex projects, when perhaps softcores are present, compile times are not very fast.

For these reasons, software simulations can be carried out.

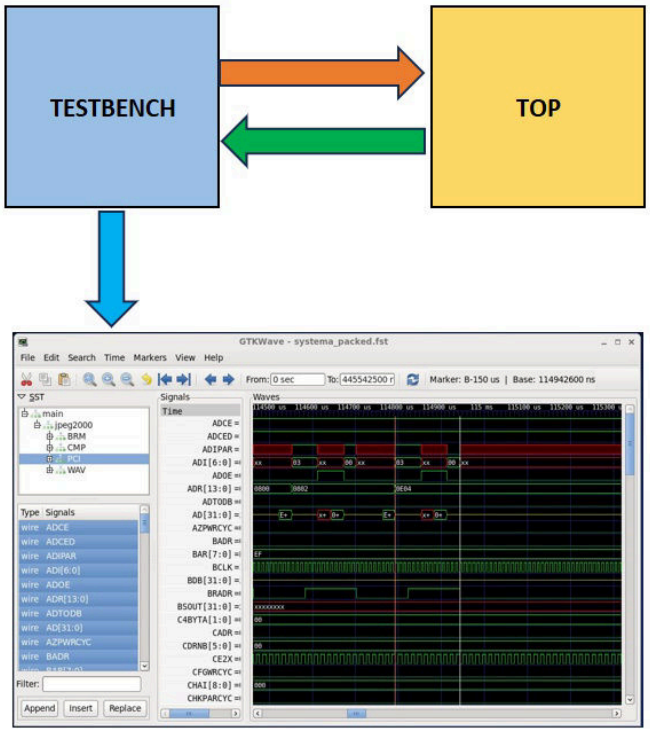
Our main module, called in jargon top, as we have seen, will have, in the declaration a number of Input/Output signals which, as already mentioned, by means of a special configuration file, we are going to associate with physical pins of the FPGA indicating also other attributes (Input, Output, Open drain etc..).

The concept of simulation in FPGAs is very simple: we create a module, in testbench jargon, that instantiates the top module and manages its I/O via software through programmed "stimuli."

There is then the option of viewing the exchanged signals in a software logic analyzer or saving them to disk and viewing them with another tool.

Some development systems have the simulator built in, others do not. However, there are free simulators, such as Icarus, that allow compilation and simulation of a Verilog project by producing a file that can be viewed by GTKWave, also a free tool.

Fields of application



Due to their versatility and increased power in recent years, FPGAs have seen increased use in architectures historically the preserve of traditional CPUs.

The first massive use was in the field of communications; with the ability to implement customized protocols and high data transfer rates, today we find FPGAs starting from "inexpensive" equipment such as modems and TVs, to military encoders/decoders, radio links and interface cards between servers.

Next, the most manned field of application is numerical signal processing (DSP). Thus, we talk about Filters, FFTs, convolutions, image analysis, noise analysis, etc.

Another application where the use of FPGAs is growing rapidly are hardware accelerators for high-performance computers and cryptanalysis. It is estimated that in the next few years FPGA-based hardware will equal that of traditional CPUs in presence.

Due to the "parallel" nature and appropriate special blocks, another field in which FPGAs are emerging strongly are neural networks and machine learning. So systems for computer vision and pattern recognition.

Finally, FPGAs are used as development systems for ASICs, chips that are architecturally very similar but are designed directly in the factory on silicon. The cost of an ASIC chip is much lower than an FPGA; they are designed for high volume applications, such as controllers for Hard Disks, graphics cards, etc. The problem is that an initial investment of a few hundred thousand is required to make the first chip, an amount that is repeated in case of revisions for errors. Being able to simulate a circuit in the field, and not just analytically, can be a significant risk reduction.

Accomplishments accessible to all

We have seen the structure of an FPGA and its parallelism, also we have learned that there are specialized blocks to increase the firepower. We have glimpsed the enormous potential that comes from an architecture that can be composed at will, now we ask: what can we accomplish?

Interface shield to commercial boards

Imagine you need ten SPI interfaces working in parallel and pre-processing data, there are no MCUs to contain them. Making them with an FPGA is quite simple, made one interface you multiply it. Same of course for UART, I2C and CAN.

Protocol managers

Another use could be a concentrator of signals from various sensors communicating with different protocols, such as 1Wire, CAN, RS485, etc. The FPGA could then communicate with an MCU or IoT gateway using a fast SPI interface.

There are so many applications, many we certainly have not thought of so far because they are impossible to achieve by traditional methods, but the ones that are certainly most intriguing are softcore.

Softcore

We have already mentioned the emblazoned softcores MicroBlaze and Nios, which are developed and supported directly by FPGA manufacturers; it is possible to find many other cores, equally complex, based on RISC-V architecture on the net because it is free, widely documented and composable, so we can both simplify and enrich it.

A softcore is less fast than a hardcore of identical architecture and internal scheme, however, by having complete control over execution, we can, for example, make realtime softcores.

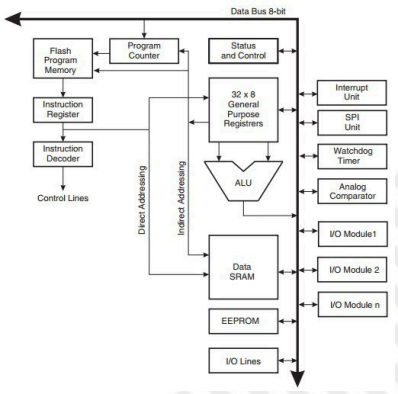
I remind you that realtime does not mean fast or performant, but deterministic. I have written an article that I encourage you to read if you want to explore this topic further.

We can reproduce existing CPUs or create new ones with arbitrary architectures and instruction sets, perhaps geared toward solving specific problems.

There is the whole retrogame world for which it is possible to find exact replicas of entire consoles, i.e., CPU, memory, and support chips. The most famous, MisTer and Mist, use hardware that is not the cheapest, but there is also NESTang, which replicates a Nintendo NES on a very inexpensive FPGA board (Tang Nano 20k).

These projects are very interesting from an educational point of view; you can see how easy it is to replicate a Z80, 6502 or 68000 in an FPGA.

Shown in the figure is the internal architecture of ATmega328p, the MCU found in Arduino that can be found as softcore on the Web.



Finally, we can think of designing micro cores to perform specific tasks and allocate a large number of them (having CLBs available) to make them all work in parallel.

We have the freedom to implement from eight-bit "Fetch-Decode-Execute" instructional cores to RISC-V or POWERPC with prefetch, cache and TMC memory. It is just a matter of available CLBs and, of course, skill.

Article published with the author's permission.

See you next issue for the second part of this very interesting article!

7800 Game Drive and Mega 7800

by Filippo Santellocco – www.santellocco.com

Since 2016, James, the British technician who makes RetroHQ products (www.retrohq.co.uk), has been thinking about a multicart for the Atari 7800 console. At the time, the situation was not the best, as the old and expensive Cuttle Cart 2 released in 2003 was not readily available.

Development began in 2019 but, due to COVID and lack of components, it was not until 2021 that James made his project a reality and announced it. Two years ago the outlook was better: multicarts Concerto and Dragonfly were available, however, whether due to some shortcomings of the latter, the advanced features gradually shown by James, or finally the reputation earned with excellent products such as Lynx Game Drive and Jaguar Game Drive, there was great anticipation among fans.
So it is natural that the first 200 units marketed in July sold out in a matter of hours. Don't worry, by the time this article is published the product will probably be available again.

But what are the features that have raised such high expectations?
Since it is a multicart, its main purpose is to load programs (games and demos), in this case placed in a microSD card. Well, the Game Drive offers full compatibility not only with classic games (including those with Pokey sound chip audio without the need to find and install one), but also with all modern homebrew productions, which are sometimes hostile to other multicarts (see Rikki & Vikki). In addition to the single Pokey chip, the cartridge also emulates double Pokey, Yamaha YM2151, BupChip, and

Game Selection 2600

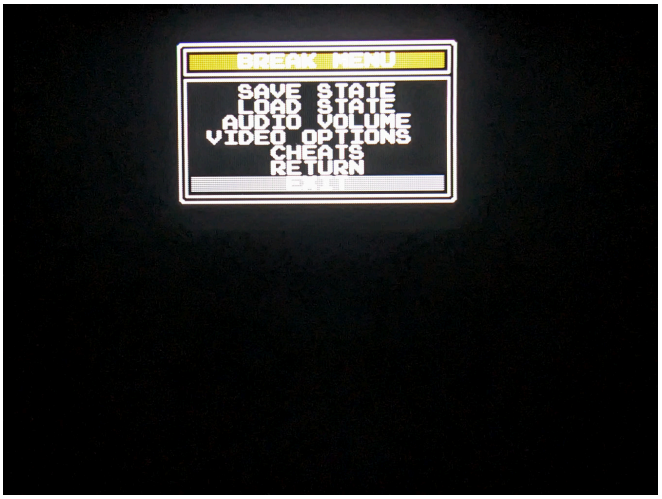
COVOX board.
Speaking of homebrew software, I open a parenthesis to point out that in recent years games have been developed that highlight the potential of a console that if marketed early in 1984 instead of 1986 would surely have been more successful. I am thinking for example of titles such as 1942, A.R.T.I., Attack of the Petscii Robots, Ghosts 'n Goblins, Keystone Koppers, Millie and Molly, PentaGo, and Popeye.

As is well known, the 7800 console is backward compatible with the Atari 2600. The Game Drive allows all its games to be loaded except those that make use of the ARM processor contained in the multicart Harmony and Melody boards. Some might turn up their noses because among the homebrews these are the most technically advanced but it must be said that 14 years after its availability, there are few VCS lovers who do not own a Harmony; the only regret is not being able to play these games on 7800 taking advantage of the RGB output. RGB output? That's right, via a cable in Sega Mega Drive II standard connected

Game Selection 7800

to the cartridge you can enjoy, simultaneously with the normal RF output, the quality assured by the RGB signal. Of course, the ideal is to use a CRT screen, but upscalers also benefit from the Game Drive's RGB.

Returning to more usual functionality for a multicart, the on-screen menu displays information about the games and allows them to be loaded. Very convenient is the ability to interrupt them and load others without having to turn off or press the buttons located on the console (currently only 7800 games). To do this just press the two buttons on the pad and move the joystick down; the game is paused and the so-called Break Menu appears. That menu provides access to various features, such as saving states (nine for each game) and cheats. In this way, even less skilled or time-sensitive players can enjoy the games all the way on the real hardware. From the menu you can also adjust the volume of individual sound chips (some games use multiple chips simultaneously), change palettes (to simulate the change that occurs when the console warms up), and enable video effects.



Break Menú

In addition, the Game Drive allows saving highscore and data in prearranged games. Finally, the ability to connect your PC to it to quickly test your programs without having to transfer them to the microSD is useful; 7800Basic, the modern language that in spite of its high level is contributing to the development of extraordinary games, is already prepared. RetroHQ has also marketed an adapter called the Mega 7800, through which Master System/Mega Drive pads and sticks and light phasers can be used. The Mega Drive pad with 6 buttons is recommended, which makes loading and saving states very fast.

In conclusion, after previewing it for two months and providing feedback to the manufacturer, I can only recommend the 7800 Game Drive. The price puts it at the high end of the market, but I think that's fair when you consider the compatibility, functionality, and presence of RGB output that would otherwise require hardware modifications that are not exactly affordable.

Front and rear views of the multicart

Nintendo Entertainment System (NES)/ Famicom - part 1

by Leonardo Miliani

The year 2023 was not only the 40th birthday of two important computers such as the Commodore 64 and the ZX Spectrum. There is another system celebrating the same years, a system whose numbers annihilate the aforementioned machines: we are talking about the "Nintendo," as it was called by the kids of the time, the console that marked the rebirth of the home video game market, reviving the industry from the deep crisis that had led big names in electronic entertainment to bankruptcy or near bankruptcy, with a boost so impressive that it represented a flywheel that then pulled all game producers toward a new creative and economic boom. The Nintendo Entertainment System (fig. 1), also known as the NES, which began as the Famicom in Japan and was later sold worldwide, is the console we are going to analyze in this article.

From playing cards

Way back in 1889, in Kyoto,Fusajiro Yamauchi founded Nintendo Karuta (also sometimes referred to as Koppai, but it is just a different reading of the same ideograms), a company devoted to the production of traditional Japanese playing cards called "hanafuda." Karuta is a transliteration of the Portuguese "paper," playing card, because the Portuguese introduced playing cards to the country in the 16th century. During the following decades Nintendo built a good reputation and business was fairly good, operating in an almost niche industry. Things change during World War II: restrictions due to the conflict and people's new needs cause the company to go through a period of crisis. In 1949/1950, due to the health condition

Fig. 1: The Nintendo Entertainment System, better known as the NES or, simply, Nintendo (author: Evan-Amos - source: Wikimedia Commons)

Fig. 2 - Hiroshi Yamauchi, the third president of Nintendo under whose leadership the NES was born (source: NYTimes.com)

of the then acting president Sekiryo Yamauchi (Yamauchi's stepson, born as Kaneda but adopted in 1907), he calls his nephew Hiroshi Yamauchi to lead the family business (fig. 2). The latter begins a plan to restructure and diversify investments in an attempt to increase Nintendo's sales. It is in the light of these changes that Nintendo makes agreements with Walt Disney to offer playing cards featuring Disney characters. The market response is immediate and positive: the new cards literally sell like hotcakes. But, as often happens at these junctures, things don't last long because Nintendo finds itself dependent on a product intended only for children as sales of adult playing cards continue to decline: this is because Japanese society at that time is changing rapidly and adults are beginning to enjoy other forms of entertainment, such as "pachinko" rooms (a type of mechanical gambling that has recently sprung up), or spending time in bowling alleys. The president tries to further diversify Nintendo's business, going so far as to invest in the instant rice market and cab companies as well, but without success.

Fig. 3 - Gunpei Yokoi, the engineer behind many Nintendo-branded hits (author: Gobonobo - source: Wikimedia Commons)

He therefore decides to create a research and development (or R&D, from Research & Development) division and hires a bright young engineer, Gunpei Yokoi, with whom to devise new entertainment systems (fig. 3).

Early electronic games

As the first toy Yokoi created the "Ultra Hand," a kind of extendable hand operating with a scissor-like handle and an extendable middle part made up of crossed segments. By squeezing the handle, the central part extends and simultaneously the hand at the other end tightens, allowing small objects (of balls, in the game package) to be grasped. It is a resounding success: introduced in 1971, some 1 million units are sold, lifting Nintendo out of the financial crisis in which it had been latent. There is also another engineer working for Nintendo, Masayuki Uemura (fig. 4): during the same period he developed a series of electronic games based on fake weapons equipped with optical sensors to detect whether the target has been hit or not, the "Beam Guns." More than 1 million of these games are also sold.

In 1974 Nintendo released one of the earliest examples of arcade games, "Wild GunMan," an electro-mechanical system based on a projector that projects images of an Old West gunslinger onto a screen. When his eyes glow, it is a sign that the gunman is about to shoot: if the player is faster than him and hits him with his fake gun, he wins the duel (the game is also released in North America in 1976 through Sega). Several children's game series based on GunMan and Beam Gun are also produced (fig. 5). Yamauchi realizes that the future is electronic video games and signs an agreement with Magnavox to distribute the Odyssey console in Japan.

Around 1977, the R&D division is split into two: R&D1, headed by Yokoi, and R&D2, headed by Uemura, are born. R&D1 begins work on a battery-powered handheld game based on a liquid crystal display, which is introduced in 1980 as "Game&Watch." It was a worldwide success: several variants were presented, each with a different game, and in its first year 14 million were sold. Some versions, based on famous arcades, do even better: the one based on "Donkey Kong" alone sells 8 million units. The Game&Watch also introduces an element that we will find again later: the cross control (fig. 6). R&D2 produces Pong-like home consoles in cooperation with Mitsubishi Electric, the first of which is the Color TV-Game 6, which is soon followed by the Color TV-Game 15 (fig. 7): both allow you to play different variants of Pong.

R&D1 released its first electronic arcade games in 1978 and 1979, "Sheriff" and "Radar Scope," respectively: the

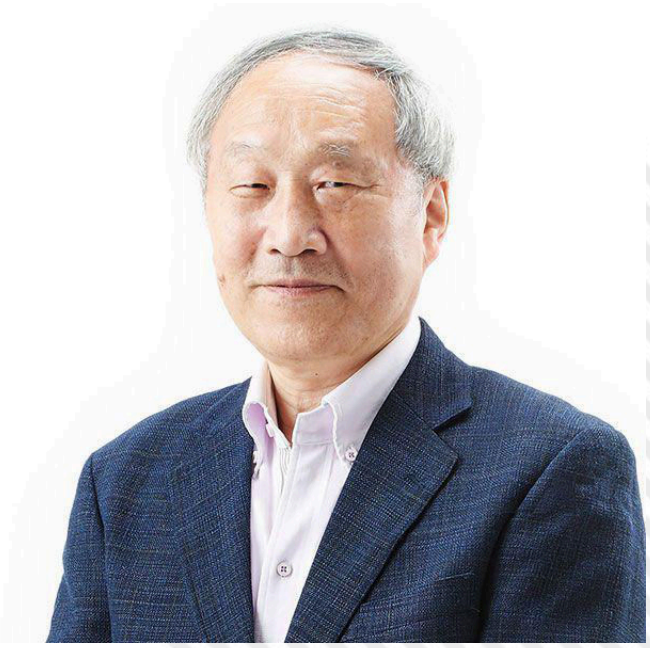


Fig. 4 - Masayuki Uemura, the creator of the NES/ Famicom (source: Memoriabit.com.br)

Fig. 5 - A version of GunMan with a Beam Gun included in the package (fote: Beforemario.com)

latter, although it managed to rival well against "Galaxian" on Japanese soil, did not establish itself outside Japan, turning out to be an overall commercial failure. To recover from the economic losses, Yokoi puts Shigeru Miyamoto to work on another arcade, which is released in 1981 and immediately becomes a worldwide success: "Donkey Kong." At the same time, in 1980, Nintendo of America was created, a subsidiary dedicated to the distribution of Nintendo arcades in North America: this is also something we will find later.

Fig. 6 - A Game & Watch: note the distinctive cross control that would later be reused in NES/Famicom controllers as well (author: Federico Leva - source: Wikimedia Commons)

Fig. 7 - The Nintendo Color TV-Game 15, a precursor to Nintendo's future programmable console (author: MagentaGreen - source: Wikimedia Commons)

Kong to the Famicom

The success of "Donkey Kong" in arcades led many people to want to play it in their homes as well. After a controversial three-way negotiation between Nintendo, Atari and Coleco, the latter won the conversion and offered the game cartridge paired with its new ColecoVision console. Success is immediate: people buy the ColecoVision because they want "Donkey Kong." Nintendo realizes that the home video game market can be a new, fruitful area in which to invest and commissions R&D2 to develop its own console. Stakes are put in place, however: it must be technologically superior to the competition when it debuts and remain so for at least a year, and it must also be cheap to produce.

Uemura begins to study the project. He initially thinks of using "Donkey Kong" hardware as the basis, which is unmatched even by the ColecoVision, although the conversion of the game is qualitatively good, but the cost of the various components, such as the arcade's Zilog Z80 CPU, do not allow for the production of an inexpensive console. Also discarded for the same reason are the 16-bit CPUs that were appearing on the market at that time, which were much more advanced but much more expensive than the 8-bit CPUs. In the end, they opt for a microprocessor that has been on the market for a few years, which is cheap, widespread, and fairly high-performance, namely the MOS 6502. In order to optimize costs, it is decided to have the CPU manufactured by Ricoh, a large manufacturer

Fig. 8 - The CPU produced by Ricoh for the Famicom/ NES, here PAL version (author: PantheraLeo1359531 - source: Wikimedia Commons)

Fig. 9 - The Nintendo Family Computer, better known as the Famicom (author: Evan-Amos - source: Wikimedia Commons)

of electronic and integrated components, with which the processor core is revised: from the 6502, the BCD calculation logic is removed and a programmable sound generator, or PSG, from "Programmable Sound Generator," is integrated (fig. 8). To aid the CPU in video management, it is joined by another integrated named PPU ("Picture Processing Unit"), capable of generating an image with a resolution of 256x240 pixels and being able to handle up to 64 different sprites-a real quantum leap over competing graphics chips. The idea of equipping the console with mass-market peripherals such as floppy disks is discarded in favor of the cheaper and more popular as well as reliable and safe ROM cartridges. To make it clear that this is a gaming device, shapes and colors are purposely chosen such that no one seeing the console can confuse it with another type of product.

The first prototype is produced in October 1982: the console is a small white plastic object with red inserts, with a large cartridge release system placed in the center of the top. This device is suggested by Yokoi, who thinks it could be a fun element for kids, and Uemura adopts the idea. The joysticks, permanently attached to the console, are red/gold in color and feature another Yokoy innovation, the D-Pad: this 4-way control device is borrowed directly from Game&Watch.

The console is ready, it just needs to be "christened" and then presented to the press. Initially the name GameCom is thought of but then it opts for Famicom, a contraction of "FAMIly COMputer," meaning a device intended for the whole family. There is one problem, however: Sharp already produces a line of electric ovens called "FamiCon" (with the final "n"). To avoid possible legal action due to possible infringement of rights to the name, on July 15, 1983, the console is introduced as the "Family Computer" (fig. 9), and this name is used in all Nintendo communications and advertisements, even though the console is known to everyone as the Famicom. Nintendo later agrees with Sharp on the term "Famicom" for exclusive use on electronic gaming systems, but "Family Computer" still remains the official name of the console, as it will always be stated on the device itself as well as on its box.

From the Famicom to the NES

Unveiled on the same day as Sega's SG-1000, the console gets off to a great start right away, aided by a sales price of 14,800 yen, which is not as low as Yamauchi desired but is still 200 yen lower than that of Sega's device. Also available with the console at the time of launch are 3 games that are nothing more than as many Nintendo arcade conversions: "Donkey Kong," "Donkey Kong Jr." and "Popeye." In the first 2 months, the Famicom sells

Fig. 10 - Vs. Dr. Mario, one of Nintendo's arcade games based on the VS. System, derived from the NES hardware (author: 17daysolderthannes - source: Wikimedia Commons)

Fig. 11 - The loading of cartridges on the NES was designed to resemble that of videotapes in epopca VCRs (author: Yagamichega- source: Wikimedia Commons)

500,000 units but slowly sales drop due to a problem: more and more consoles are sent for repair due to the fact that with some games the console will not start. A manufacturing defect is found afflicting the motherboard. Yamauchi makes a very strong decision: although the operation is financially costly, it is decided to replace all defective motherboards under warranty. This serves, according to his thinking, to create towards the company a reputation of seriousness and reliability, and the public appreciates and repays Nintendo's effort: the Famicom resumes grinding out respectable numbers becoming, by the end of 1984, the best-selling console in Japan. By early 1985 the console had sold 2.5 million units: such success prompted Yamauchi to decide to market the console in the United States as well. That market, although potentially lucrative, is very risky: North America has just faced what is considered the biggest crisis in the video game industry, with the bankruptcy of many companies and the withdrawal of several systems from the market. There is one problem, however: Nintendo of America had already tried the year before to introduce the Famicom to the U.S. market, first by seeking a deal with Atari (blown over the "Donkey Kong" conversion story), and then by trying to place it directly in stores but retailers wouldn't hear of it as a game console. Then again, it is 1983 and the video game "bang" is still making noise. Not wanting to give up on that market, and seeing how, despite the crisis in home consoles, the market for bar cabinets is still doing well, Nintendo decided to develop an arcade platform based on the Famicom's hardware, the "Nintendo VS. System": this system uses the same CPU as the Famicom to which a kit consisting of a ROM and a dedicated PPU is added, depending on the game (fig. 10). The VS. System quickly becomes the best-selling arcade system and remains in production for several years, totaling the considerable number of 100,000 units sold at the end of its commercial life. This is important because the VS. System is, for all intents and purposes, the first debut, in 1984, of the Famicom on American soil.

As sales of the Famicom and the VS. System are proceeding apace, Uemura is tasked with studying a way to make the Famicom saleable in North America. Since the video game

crisis has actually created a revulsion toward game consoles, it is decided to change the device's dress: away with the flashy colored plastics, away with the cartridge loading on the top. The device should not remind one of a game console but of a common household appliance: and for this the shape of a... video recorder is chosen! Yes, Uemura decides to disguise the console from an electronic device that was becoming very popular in households in those years. Even the front-loading of cartridges mimics exactly the insertion of a videotape into a VCR: the user must have the impression of using an ordinary electronic device and not a game console. Nintendo of America's style center helps in the development of the case so as to make something appealing to the tastes of North Americans: the two shades of gray and the squarish shape serve precisely to complement the "feel" of the household appliance (fig. 11).

It is also decided to drop the name of "Family Computer" and opt for "Nintendo System," so as not to invoke a connection to a computer or console in any way and to avoid these incriminating terms. It is later decided to insert the word "Entertainment" in the middle of the name, so as to get across the message that the device is for "entertainment" and not something to play with. And the names are carefully chosen for the various components as well: the central body of the console is named "Control Deck" while the game cartridges take on the name "Game Pak," the only reminder of the device's ludic origin.

The Nintendo Entertainment System, abbreviated to "NES," is unveiled at the Consumer Electronic Show held in June 1985. It is not put on the market, however, until several months later, and in zones: it officially launches on October 18 from New York and then, in early 1986, from Los Angeles, while distribution throughout the United States does not start until September 27, 1986.

The launch in the rest of the world is also by zones and with different distributors. The first zone includes Scandinavia and most of continental Europe such as France, Germany, and Spain, where it begins in late 1986. For the second zone, which includes Italy, Ireland, Great Britain, Australia, New Zealand, and India, distribution begins in 1987 and is mostly taken over by Mattel. The NES then arrives in South Korea, but not until 1989, and even in 1993, about 2 years before its discontinuation, in Brazil and South Africa.

Commercial life

In North America, the main market, the NES is offered at launch in the "Deluxe Set" only, which costs $179.99 and includes the "R.O.B." robot, which will only be used in 2 games including "Gyromite" included in the package, and the "Zapper" optical pistol, which will be used in about 20 games including "Duck Hunter" included with the console. Then the console alone goes on sale for $89.99, or, if you opt for the version with the dual cartridge of "Super Mario Bros." and "Duck Hunt," for $99.99.

The power of the console as well as the quality of its games are both factors driving sales of the NES: it is estimated that in 1988 alone, within a year of the start of distribution throughout the U.S., the NES sold more than 7 million units, the same number that the Commodore 64 did in its first 5 years of life. By 1990 the NES was in 30 percent of American homes and the Famicom in 37 percent of Japanese homes.

In the late 1980s and early 1990s, the first 16-bit consoles began to appear, including the TurboGrafx-16 and the Sega Mega Drive. To keep up with the times, Nintendo updated the NES and introduced the Super NES, or Super Nintendo Entertainment System, in late 1990. Despite the competition from these technically superior machines, the NES holds its own and continues to sell very well, thanks in part to the range of games that, as time goes on, squeeze more and more out of the console's capabilities. Nintendo is forced to put the Super NES on sale for only $10 more than the NES so that the small price difference between the 8-bit and 16-bit systems will entice people to buy the latter.

The NES remains on the market until 1995, when Nintendo decides to take it off the market. The Famicom, on the other hand, has a much longer life: it remains on sale in Japan until 2003 and is taken off the market because the scarcity of spare parts now makes it impossible to repair any failed units.

The NES/Famicom was the first game console to set an incredible sales record: an estimated nearly 62 million units were sold, including 42.5 million NES units. It was also one of the longest-lived, having remained on the market (in Japan) for 20 consecutive years.

That's all for this issue friends, see you in the next issue for the second and final part of the article. (Note by FF).

Fortran on Commodore 64

by Francesco Fiorentini

Those of you who follow my articles on RetroMagazine World know that I prefer to review applications and languages, although from time to time I don't disdain to try and comment on games as well. In this, my umpteenth article, however, I want to stay true to my passion by telling you about a programming language: Fortran.

Fortran is one of the earliest programming languages to have been developed; its design dates as far back as the first half of the 1950s, when, in late 1953, **John W. Backus** proposed to his superiors at IBM that they develop a more practical alternative to the assembly language for programming the IBM 704 mainframe computer.

Exactly 70 years have passed since then, but this language is still used by the scientific and engineering communities; of course, in all these years the language has undergone several modifications to try to adapt to the renewed needs of the market and developers.

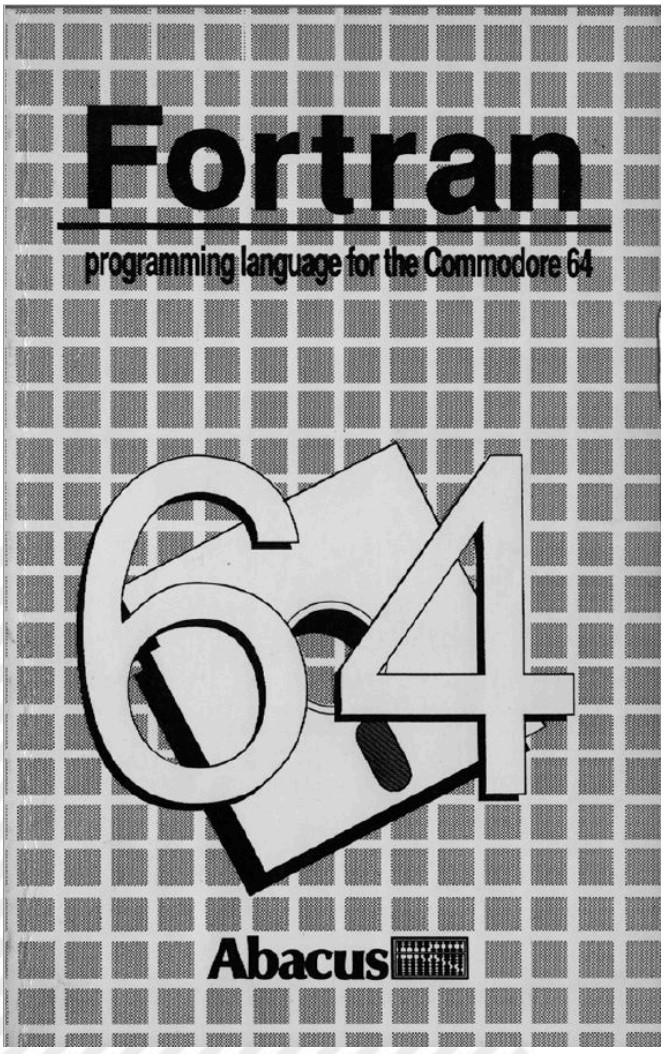


Fig. 1 - Abacus Fortran-64

Fortran and the Commodore 64

As we have already experienced, the Commodore 64 hosted all kinds of software, recreational and otherwise, and obviously, in the list of programming languages available for this home computer, Fortran could not be missing.

From my research there appear to be at least two versions of Fortran available for Commodore's best-selling 8-bit, the **Nevada Fortran** and the **Abacus Fortran-64**. My choice, for reasons of practicality, fell precisely on the latter, since the Nevada Fortran requires the CP/M cartridge to run. I therefore assumed that the Abacus product would be more accessible to those who wanted to try it on a real machine. This is not to say that I will not also do a test of Nevada Fortran in the future.

Ah, anyone who knows of other versions of Fortran please let me know. I love getting my hands on semi-unknown software. :-)

Abacus Fortran-64

The Abacus Fortran language was released in 1988 by Bob Stovere and Tim Adams. The package consists of a disk containing the compiler, linker, libraries, a handy translator (we will see later what it is used for) and a manual which, in perfect Abacus style of the period, is stripped down to the essentials.

Once Fortran-64 is loaded we will be greeted by a menu that contains all the options for creating our first program in Fortran, compiling it, linking it, and generating an executable. What immediately jumps out at us, however, is the lack of a text editor to create the source file: so how to do it?

The source file can be created in two different ways. The first is to use any word processor that produces sequential ASCII output files, and the second is to use the built-in editor in BASIC V2. Using the F2 - Back to Basic option, we will return to the BASIC prompt and can begin writing code. Each line must be entered as in BASIC, but with

Fig. 2 - Loading Abacus Fortran-64

one small exception: the colon (:) must appear in the line between the line number and the Fortran instruction. After the program has been written it can be saved to disk with BASIC's classic SAVE command.

At this point, returning to the Abacus Fortran-64 menu via the SYS49152 command (warning, read the note below), we can proceed to transform the BASIC PRG file into a SEQ (sequential) file via the F6 - Translator command.
Please note: To allow reentry into Fortran-64 without reloading, you must run a SYS49152 immediately after returning to BASIC. After saving the file, you can run a second SYS49152 to return to the main menu.

Our first program in Fortran-64

Given the possibility of using the editor in Basic, I chose this very option to write my first program in Fortran with the Commodore 64.
Once back in Basic, I then typed SYS49152 and after that I wrote the following program:

```
10 : program test
15 :* this program will write a
16 :* list of the odd numbers
16 :* from 1 to 10 to screen
20 : integer a,b
30 : open 3.3
40 : do 10 i = 1, 10, 2
50 : write (3,100) i
60 : 10 continuous
65 : close 3
70 : stop
80 : 100 format(i5)
90 : end
```

and then saved it with the command: SAVE "ODDNUM",8

Next, I typed SYS49152 again to return to the Abacus Fortran-64 menu and transformed it into a sequential file using the F6 - Translator command.
The Translator program asks us for the name of the file we created and saved in the BASIC environment:
enter program file name: oddnum
Then we are asked under what name we want to save the sequential file:
enter sequential file name: oddseq
At this point we are ready to compile our program.

Now let's compile it...

From the main menu we will choose the item F1 - Compiler. The compiler will ask us to provide the name of our program, that is, the sequential file we created with the translator, and then to provide the name of the object file we want to create:

```
enter source file name: oddseq
enter object file name: oddobj2
```

Fig. 3 - The Main Menu

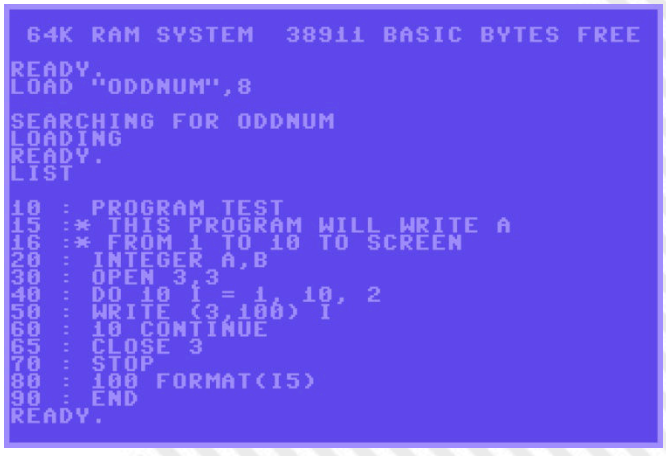


Fig. 4 - The program just typed in

Fig. 5 - The program 'translated' with Translator

At this point the compilation phase of the program will begin and any errors that the compiler will have detected will be reported on the screen. I found this option really interesting because each error is indicated exactly under the line that contains it, definitely making the programmer's life easier.

Errors will be reported using the following format:
***** ERROR xxx yyy ****
xxx = error code
yyy = position of the character in the line that caused the error (excluding spaces)
For the list of 64 error codes, see the manual on page 47: Appendix D - Error Description.

All done? Of course not, there is still one crucial step missing before we can test the execution of our program. Using the linker we will have to link the object code, that is, the translation of the source code into machine language, with the language libraries needed to be able to run the program.
...and finally the Linker!

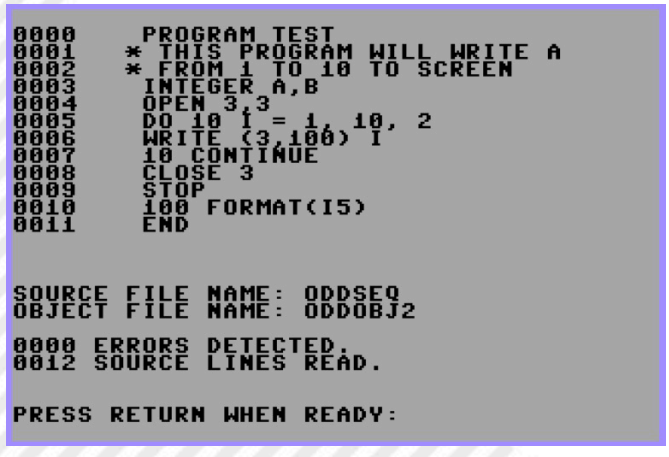


Fig. 6 - The compilation process

To use the linker all we have to do is choose F3 - Linker from the main menu and then communicate the name of our object program, in our case ODDOBJ2.

The linker will automatically link all the modules needed to execute our code, and these will also be displayed if we choose. Finally, the linker will inform us at what address our executable code is located:
modules loaded:
test $0850
clwbuf $0903
set-up $0915
in2as $09d3
fiwbuf $0abf
wformt $0c2d
faterr $11f6

runable code resides
from $0850 to $1230

At the end of the linking process, the executable program will be resident in memory. Using the BASIC LIST command, we will see on the first line a REM command with the name of the Fortran program, and on the second line a SYS command with the execution address of the program. The rest of the program is in machine language. At this point the program can be saved to disk or run with the BASIC RUN command.

The output of the LIST command:
10 rem program test
20 sys02128
Note that the SYS points to address 2128, in hexadecimal $850. Yes, the very same starting address given by the linker.

A look at the code

10 : program test
Simply the name of the program

15 :* this program will write a
16 :* list of the odd numbers
16 :* from 1 to 10 to screen
Comment lines

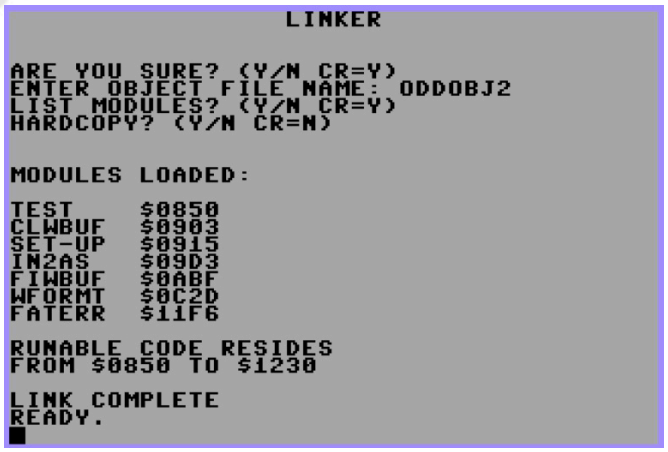


Fig. 7 - The linked program

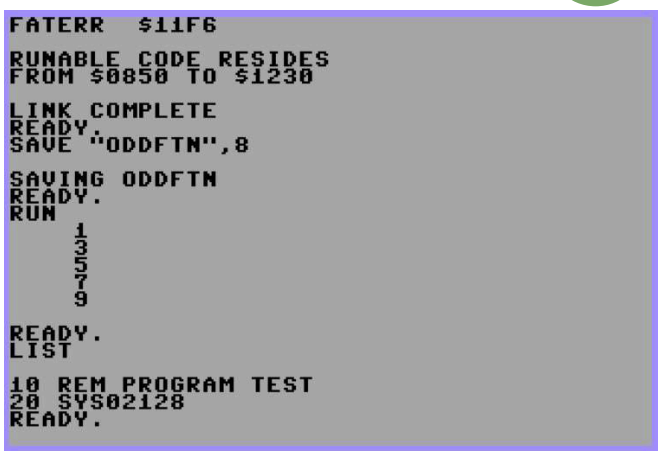


Fig. 8 - The running program

20 : integer a,b

Defines two variables of integer type (not used)

30 : open 3.3

It opens a connection with a device. I personally referred to this table to identify the video:

Device number 0 | keyboard (not counted as first device!).

Device number 1 | datasette | secondary number 0 = for loading; 1 = for writing; 2 = for writing a file followed by an EOT (End Of Tape) sign

Device number 2 | modem / RS-232 | secondary number 0

Device number 3 | screen | secondary number 0 or 1

Device number 4 or 5 | printer | secondary number 0 = uppercase letters/graphic symbols 7 = uppercase letters/ lowercase letters

Device number 8 to 15 | disk drives | secondary number 2-14 = data channels[1]; 15 channel for commands

40 : do 10 i = 1, 10, 2

DO label varnam = is,if [,ic]

Where: label is an unsigned integer

varnam is a simple variable

is is a constant or an integer variable (initial value)

if is a constant or integer variable (final value)

ic is a signed integer constant (increment)

DO cycle from 1 to 10 with step 2

50 : write (3,100) i

Print variable i (formatted on line 80) on the screen.

60 : 10 continuous

End of the DO loop

65 : close 3

Closes the connection with the device (see line 30)

70 : stop

Terminate program execution and return to Basic

80 : 100 format(i5)

Format the printout of a 5-digit integer

90 : end

Indicates the end of a program unit.

Now it's your turn! Try writing your first program and compile it the same way.

Have fun!

Useful Links

Download Abacus Fortran-64 and its manual:
- https://www.lyonlabs.org/commodore/onrequest/collections.html

Pagina WIKI on Fortran:
- https://it.wikipedia.org/wiki/Fortran

Playing with Infinite Lives - The Ultimate Challenge

by Marco Pistorio

Greetings to all readers of RetroMagazine World. I am returning to this topic again because I think it is worthwhile and of interest to many "buddy crackers" :)

I suggest that you re-read the previous articles dealing with this issue, if you are not familiar with them, because in this article I am going to develop some elements that I have already discussed in the very previous articles, published respectively in issue 03-IT of RetroMagazine World ("Endless Lives with the Commodore 64") and issue 01-EN ("Playing Endless Lives on the C64 - "The Challenge Continues").

Here's a short recap. I have presented a tool that allows you to examine different snapshots of the C64's memory and facilitates finding the location containing the number of lives available within a generic game.

I then showed how, by means of NOP instructions, to disable the change in the content of the identified location, which would otherwise normally be done by a decrement. Finally, after modifying the game code with the appropriate changes, I showed how to save everything to disk and thus obtain a "customized" version of the modified game. This is an avenue that is, in fact, viable. However, very often a "trainer," a program is created to achieve a similar result.

We will therefore focus on this avenue and try to understand what exactly a trainer is, how it operates, and what benefits its use brings.

THE TRAINER

A trainer is an aid, a simple program that modifies for us the code of the game for which it is made and allows us, for example, to play infinite lives.

What is the advantage of using a trainer? The advantage is essentially that it loads the game and leaves the user with the choice of playing the game in normal mode, or playing in 'facilitated' mode, e.g., avoiding losing available lives during the game.

The advantage then seems pretty clear to me.

But how do you make a trainer?

First, one must be clear about what aspect of the game under consideration is to be modified. Usually it is just the ability to play with infinite lives, but in other cases a trainer allows you to modify other behaviors or details of the game. The number of swords, resources available during the game, or the speed of gameplay, or even the ability to end a level and move on to the next one immediately.

If we intend to work only on infinite lives, everything we have already seen in the previous two articles remains valid to arrive at determining the memory location that holds that information during the execution of the game. But we will not directly change the game code nor save such a modified copy of the game itself.

Instead, we will write a trainer.

The game will NOT start from its initial sys, but from our trainer. The trainer will ask us the question, "Do you want to play infinite lives?" and if we answer "No," the trainer will do absolutely nothing, it will terminate and start the game.

If we answer "Yes" instead, the appropriate change of the game code will be made from within the trainer, then the trainer will end and, when the game starts, we will have infinite lives!

IMPORTANT CONSIDERATIONS

The trainer will occupy part of the computer's memory, and to act, it must be present along with the game code that it will, eventually, go to modify.

Fortunately, trainers usually occupy little memory.

Another important concept, already expressed for the truth, is that the game cannot start directly, but will have to be launched by the trainer. Only then will we have the choice of playing the game without any modification, or playing it with some "facilitator" (e.g. infinite lives).

If it is chosen to play without any modification, the game code must remain unchanged.

ANATOMY OF A REAL TRAINER

I have recently made, under the signature of the group "HOKUTO FORCE" (which I salute, taking advantage of these lines) a trainer for the game "TSI_MAZE_MAN". My trainer acts on as many as 5 aspects of this game.

I show you here a simplified version of it that only allows you to play endless lives with this game or not.

```
.pc=$0801 "basic starter"
:BasicUpstart(start)

.pc=$7A00 "start code trainers"
```

```
start:

         lda #0
         sta $d020
         sta $d021

////////////////////////////////////////////////////////
/////

        ldx #0
su_0:
         lda testo0,x
         jsr $ffd2
         beq cont
         inx
         jmp su_0

cont:

////////////////////////////////////////////////////////
/////
        ldx #0
su_1:
         lda testo1,x
         beq leggi_tasto_premuto_1
         jsr $ffd2
         inx
         jmp su_1

leggi_tasto_premuto_1:
         jsr $FFE4

         cmp #89                          //Y
        beq si_1
         cmp #78                          //N
         beq no_oppure_return_1
         cmp #13                          //RETURN
         beq no_oppure_return_1
         jmp leggi_tasto_premuto_1

si_1:
         jsr $ffd2

         lda #$EA                  //opcode NOP
         sta $832c
         sta $832d
         sta $832e

         sta $832f
         sta $8330

         lda #13
         jsr $ffd2
        jmp continua_1

no_oppure_return_1:
         lda #78
         jsr $ffd2
         lda #13
         jsr $ffd2
        jmp continua_1

////////////////////////////////////////////////////////
//////

continua_1:

         jmp $7FE0

testo0:
.byte 147,05
.text "----------------------------------------"
.text "TSI MAZE MAN (C) 1983 TRAINED BY MP SOFT"
.text "----------------------------------------"
.byte 13
.byte 13,158
.byte 0

testo1:
.text "UNLIMITED LIVES <Y/N> "
.byte 0

.pc=$7FE0 "start dump game data"
.import binary "dump2.bin",2
```

This is a few lines of code, with KICK ASSEMBLER syntax. Immediately after setting the black color for screen and background, simple text is written on the screen introducing the game and this trainer of his, "TSI MAZE MAN (C) 1983 TRAINED BY MP SOFT"

Next, the fateful question is asked, again on the screen, "UNLIMITED LIVES <Y/N> " and the choice made by the user is checked.

By exploiting the kernal $FFE4 routine we are able to know which key is pressed.

If N was pressed or return was beaten, the key pressed is written on the screen (N applies in both cases), then we give a return and finally the game starts.

If, on the other hand, Y was pressed, the trainer comes into operation by modifying the appropriate memory locations, that is, by loading the value #$EA (which corresponds to good old "NOP") into the accumulator and then writing it into locations $832C, $832D, $832E, $832F, $8330.

Instead, leaving such locations at their original values would decrease lives normally during the game.

Again, we give a return on the screen (with the instructions lda #13 and jsr $ffd2) and finally the game starts (with jmp $7FE0).

I know this last part is tough but seeing the trainer code was a must :)

A warm greeting to all RetroMagazine World readers and see you next time!

For those who are curious, I provide below the link to the TSI MAZE MAN game with my "real" trainer:

- https://csdb.dk/release/?id=235283

Weird products
(C64 – for beginners)

by Eugenio Rapella

Francis is learning multiplication between fractions. Ivy, the teacher, notices that Francis has been writing:

$$\frac{1}{4}\cdot\frac{8}{5}=\frac{18}{45}$$

"You can't perform multiplication this way, side-by-side the digits," says Ivy; Francis argues that the operation is correct: simplifying, in both members gives 2/5.
The "question" comes from the book "Professor Stewart's Cabinet of Mathematical Curiosities"; the author invites the reader to find for which combinations of digits (zero is not considered in this problem) the "Francis method" works, apart from trivial situations in which the numerator and denominator of each of the two fractions are equal to each other. In the solutions chapter, Ian points out that there are exactly 14 solutions (there are seven choices for the four digits plus the other seven that are obtained ... by flipping the fractions).
Can you find them even without the help of our trusty C64?

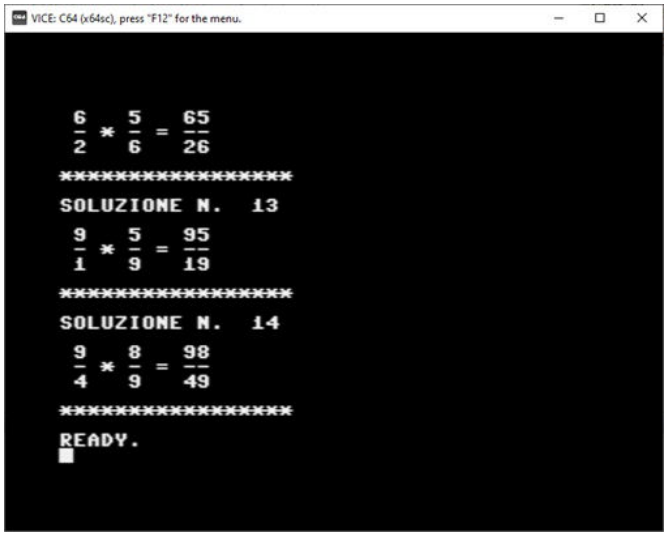
My Commodore 64 was almost offended by the simplicity of the question, but I explained that it was a matter of stretching it before engaging it in more complex tasks.

Here then is how he addressed the problem:

```
100 poke 53280,0:poke 53281,0:print chr$(5)
110 for a=1 to 9:for b=1 to 9
120 for c=1 to 9:for d=1 to 9
130 if a=b and c=d then 220
140 n=10*b+d:m=10*a+c
150 if a*c*n<>b*d*m then 220
160 k=k+1:print:print"soluzione n. ";k:print
170 print a;" ";c;" ";m
180 print" - * - = --"
190 print b;" "d;" ";n
200 print:print"*****************"
220 next:next:next:next
```

If A and B are the numerator and denominator of the first fraction and C and D are the numerator and denominator of the second, the "Francis method" is successful if the fraction (A*C)/(B*D) has the same value as (10*A+C)/(10*B+D): "flanking" the digit A with the digit C yields the number 10*A+C; the same is true for the denominator.
Here then, after the ritual colorings (at 100), no less than four "for-next" cycles (instr. 110 and 120) open and close at 220.
The C64 has the luxury of assaying all 9*9*9*9=6561 possibilities: at 130 you check whether it is a trivial solution (in which case you go straight to 220).
Otherwise, at 140 we calculate numerator M and denominator N of the fraction at the second member. If (A*C)/(B*D) is different from M/N, i.e., if A*C*N<>B*D*M (instr. 150) the "Francis method" fails and you go to 220, otherwise you have found one of the 14 solutions that is printed.

Here is the screenshot at the end of the work:



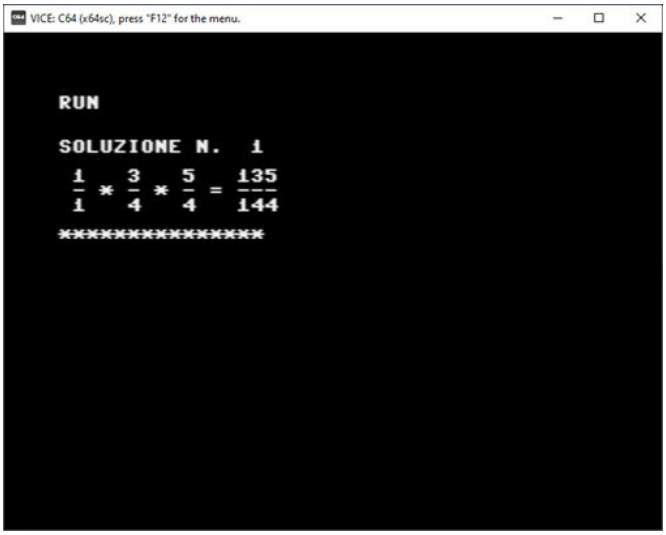
Will Francis' method also work with three fractions? The previous program can be easily modified:

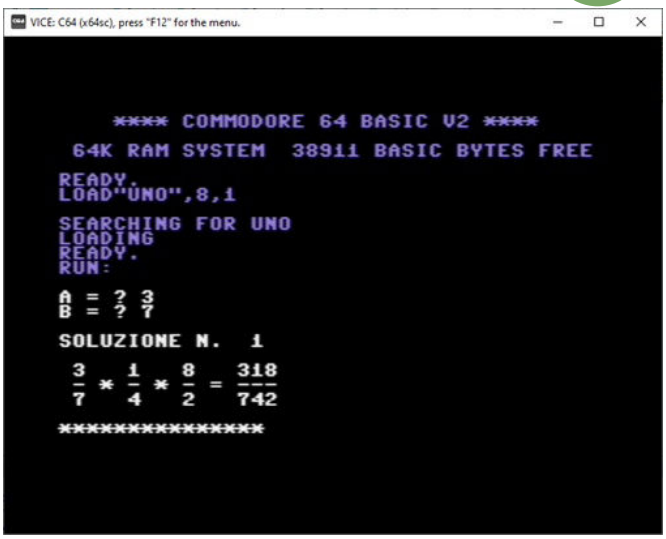
```
100 poke 53280,0:poke 53281,0:print chr$(5)
110 for a=1 to 9:for b=1 to 9
120 for c=1 to 9:for d=1 to 9
```

```
130 for e=1 to 9:for f=1 to 9
140 if a=b and c=d and e=f then 230
150 n=100*a+10*c+e:m=100*b+10*d+f
160 if a*c*e*m<>b*d*f*n then 230
170 k=k+1:print:print"soluzione n. ";k:print
180 print a;" ";c;" ";e;" ";n
190 print" - * - * - = ---"
200 print b;" ";d;" ";f;" ";m
210 print:print"***************"
230 next:next:next:next:next:next
```

but as many as six nested cycles from 1 to 9 means a beauty of 9^6=531441 iterations and impractical waiting times.

However, here is the first solution found by the C64 and a solution obtained by manually setting the values of A and B (just to check for other solutions without having to wait an eternity).





Actually, I tried to modify the program to see if our C64 could find some solutions even with four fractions--yes, yes, there are, here are some:

$$\frac{2}{1}\cdot\frac{2}{8}\cdot\frac{8}{2}\cdot\frac{5}{8}=\frac{2285}{1828}$$

$$\frac{2}{1}\cdot\frac{3}{3}\cdot\frac{1}{2}\cdot\frac{7}{4}=\frac{2317}{1324}$$

$$\frac{2}{1}\cdot\frac{3}{8}\cdot\frac{6}{9}\cdot\frac{5}{2}=\frac{2365}{1892}$$

And bravo to our Commodore 64! It really deserves it.

Permutations (for C64)

by Eugenio Rapella

The letters that form the "word" EFBDCA are the first six letters of the alphabet. What is the word, again formed by these letters, that is immediately following it in alphabetical order? Given what is required, we will try, as far as possible, to keep the first letters unchanged so ... we will start at the bottom.

The last two letters are CA; since C follows A in alphabetical order, if we switched them, we would have a word (EFBDAC) preceding the one under consideration. The same is true for the DC pair. We continue upward by examining contiguous pairs of letters; now it is the turn of BD and this time, since B precedes D we will have the possibility of finding a letter to the right of B that can take its place (at the limit it will be D itself).

Of course, if the source word were FEDCBA, we would not find the appropriate pair since this word is the last in alphabetical order formed by the letters under consideration. With whom should B be exchanged? From all those to the right of B, i.e. DCA, we will choose the one that, in alphabetical order, follows it first. Since these are already arranged in reverse alphabetical order, we will simply start with the last one and work our way up until we find the first one that follows ours in alphabetical order. A does not fit (A precedes B), but C passes the test so we will switch places between C and B. For now we have EFC followed by DBA. We are at the end of the line: we are left to arrange the letters after C so that the "final queue" is ... in alphabetical order. The swap we made did not change the alphabetical order of this "queue," which turns out to be in reverse alphabetical order. It will suffice simply to reverse it: DBA > ABD. Moral: the anagram of ABCDEF that follows EFBDCA in alphabetical order is EFCABD.

The whole thing looks a bit complicated, but that is only appearance. Let's review the steps more succinctly with a new example: what is the anagram of "FOOTNOTES" that follows this word in alphabetical order? We use the symbol "<" for "precedes" and ">" for "follows." We have C>A<L<Z>O>N>I. Going back up we find that the letter to be interchanged is "L." Among the letters to the right of L, which one follows it first? They are already arranged in descending order, so starting from the bottom, I>L? No. N>L? Yes! We exchange L for N: CAN-ZOLI and now reverse the order of the last four letters: CANILOZ. In alphabetical order, the anagram immediately following CALZONI is CANILOZ.

Here is how our trusty C64 addresses and solves the problem:

```
100 input "write a word";n$:d=len(n$):dim a(d)
110 for k=1 to d:a(k)=asc(mid$(n$,k,1)):next
120 print "press a key to start and"
130 print "to go to next"
140 get r$: if r$="" then 140
150 for k=1 to d:print chr$(a(k));:next:print
160 i=d-1
170 if a(i)>=a(i+1) then i=i-1:goto170
180 if i=0 then print:print "end":end
190 j=d
200 if a(j)<=a(i) then j=j-1:goto200
210 t=a(j):a(j)=a(i):a(i)=t
220 i=i+1:j=d
230 if i<j then t=a(j): a(j)=a(i):
a(i)=t:i=i+1:j=j-1: goto 230
240 goto 140
```

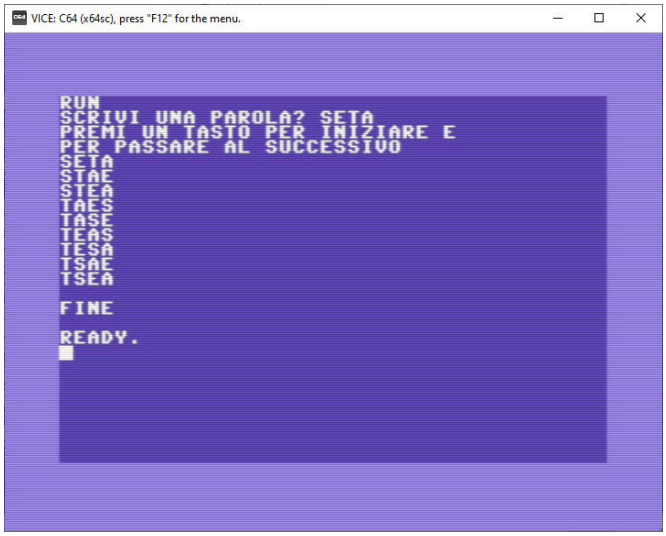
The ASCII codes of the letters that form the word, required at 100, are loaded into the vector A(..); the size of A(..) is D, equal to the length of the word (if N$="hook", it is D=3 and A=(65,77,79)).

At 150 the letters now contained in A(..) are printed; the actual program begins at 160.

Beginning at the end, pairs of contiguous letters are compared until the first pair is found for which A(I) precedes A(I+1) (instr. 170): A(I) will be the letter to be exchanged. If the anagram is the last in alphabetical order, the variable I reaches the value 0 and the program ends. At instruction 190-200, we start again from the bottom looking for the first letter A(J) following A(I): A(J) will be the letter to be exchanged for A(I).

At 210 we swap and at 220-230 we reverse the queue order; at 240 we return to 140 to print and search for the next anagram.

Here is an example of its use:



But ... what can such a program be used for? Not much ... however, if the initial word consists of the first N letters in alphabetical order (e.g. "abc" for N=3), the C64 gives us all the possible anagrams, i.e., the list of permutations.

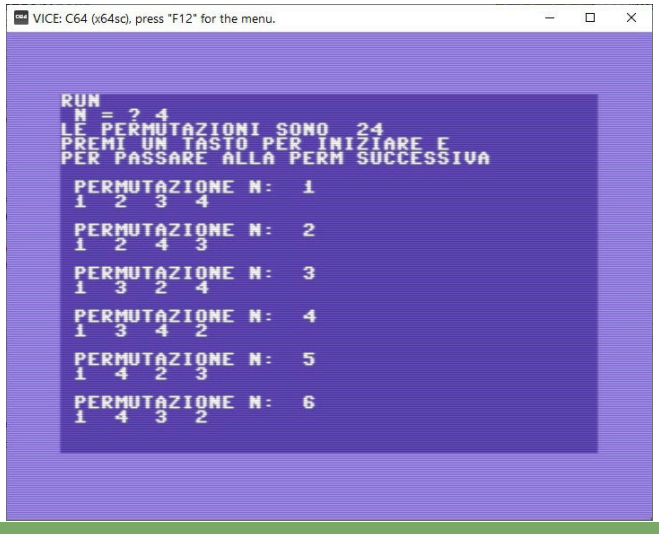
Let's abandon the literal example and reformulate the program using the numbers 1 to N. Here's how:

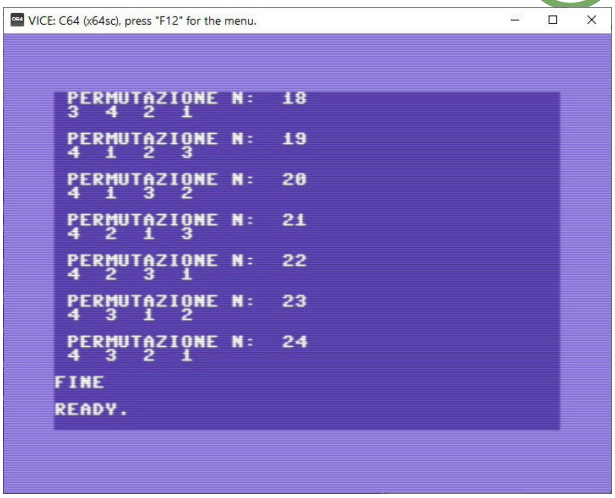
```
100 input" n = ";n:dim a(n):f=1:c=0
110 for k=1 to n:a(k)=k:f=f*k:next
120 print "permutations are ";f
130 print "press a key to start and"
140 print "to go to the next perm"
150 get r$: if r$="" then 150
160 c=c+1
170 print:print" permutation n: ";c
180 for k=1 to n:print a(k);:next:print
190 i=n-1
200 if a(i)>=a(i+1) then i=i-1:goto200
210 if i=0 then print:print "end":end
220 j=n
230 if a(j)<=a(i) then j=j-1:goto230
240 t=a(j):a(j)=a(i):a(i)=t
250 i=i+1:j=n
260 if i<j then
t=a(j):a(j)=a(i):a(i)=t:i=i+1:j=j-1:goto 260
270 goto 150
```

After requesting N, our C64 loads the vector A(..) with the numbers from 1 to N and takes the opportunity to calculate the factorial of N (F=N!=1×2×3×..×N) which is the number of permutations it will have to generate.

The rest is identical to the previous program, the variable C serving as a counter of the various permutations.

Here is the initial screen shot and the final screen shot of a usage example:





Of course, the code can be used as a subroutine in programs where it is required to generate all permutations of N objects. Here then is a question I had proposed in a test for third-year high school students:

Permuting the digits 1,2,3,4,5 in all possible ways yields 120 natural numbers. What is the sum of these 120 numbers? Obviously students (and RMW readers) are required to solve the problem with a ... shortcut, but the C64 can afford to perform the sum in full. Get to work, then compare your solution with that obtained from the program below where the permutation contained in the vector A(..) is transformed into a number in instruction 130. Each digit contributes to the creation of the number V depending on its position in the vector A(..): A(1) "digit of units," A(2) "digit of tens," and so on. Thus, if A=(2,5,1,4,3), we will have:

V=2*10^0+5*10^1+1*10^2+4*10^3+3*10^4=
=2+50+100+4000+30000=34152

At 140 in the variable S, the sum of the various values is updated so that, in the end, S contains the solution.

```
100 n=5:dim a(n):c=0
110 for k=1 to n:a(k)=k:next
120 c=c+1:v=0
130 for k=1 to n:v=v+a(k)*10^(k-1):next
140 print v; "addend n.";c:s=s+v
150 i=n-1
160 if a(i)>=a(i+1) then i=i-1:goto160
170 if i=0 then print:print "total: ";s:end
180 j=n
190 if a(j)<=a(i) then j=j-1:goto190
200 t=a(j):a(j)=a(i):a(i)=t
210 i=i+1:j=n
220 if i<j then
t=a(j):a(j)=a(i):a(i)=t:i=i+1:j=j-1:goto 220
230 goto 120
```

I would say that our Commodore has certainly passed the test, what do you say?

Libdragon Development: the kit for creating games on Nintendo 64 - part 1

by Takahiro Yoshioka – translation: Carlo Nithaiah Del Mar Pirazzini

Libdragon is a development kit (SDK) for making games on Nintendo 64.
It allows development using the C program language and the GCC compiler up to C11.
For novice users, it can be very intimidating and difficult to use, so a thorough knowledge of C is recommended before embarking on the development journey on N64 through Libdragon.
In this article we will look at how to start preparing our workstation.
This guide is the bare minimum for making a Libdragon-based game. We will only have a text interface, which is unexciting, but it is a necessary stepping stone to creating more complex projects in the future.

Installation of Libdragon

There are many ways to start developing on Pc. A more complete guide may be the one on the official wiki (https://github.com/DragonMinded/libdragon/wiki/Installing-libdragon), but I will try to provide a step-by-step approach of configuring Libdragon while keeping a "simple" style. Since I use Windows as my platform, I will show the process with this OS. For other platforms, I invite you to consult the same wiki I left above. However, the process should be quite similar.

DOCKER METHOD

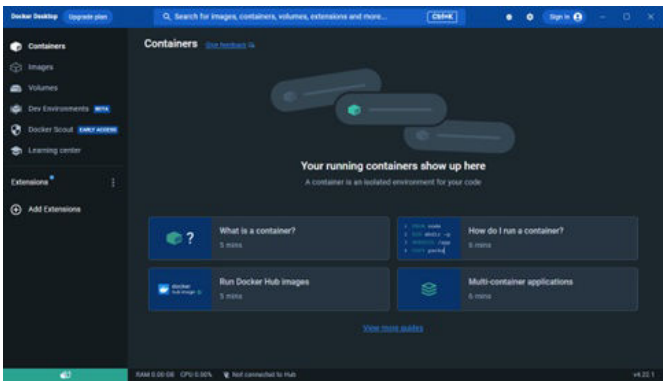
A method recommended by the development team that consists of using Docker, a software that allows "containers," small, self-enclosed virtual machines, to run. It is used to run programs that run consistently on different platforms because the actual environment on which it runs is virtualized.

STEP: 1 Docker Installation

We can download Docker from this address https://www.docker.com/ and install it.
Docker can be run as a command-line application, but Docker Desktop can be useful because of a more intuitive interface.

Some things to know:
- On Windows, you may need to upgrade the Windows Subsystem for Linux (WSL) to the latest version.
- If you mount a VPN, it could affect and interfere with Docker.

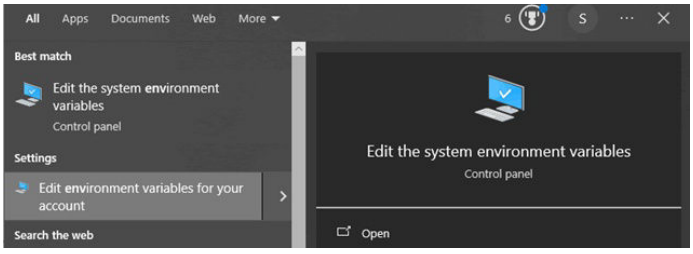
That said, let's start
The first screen that will come up will be this:



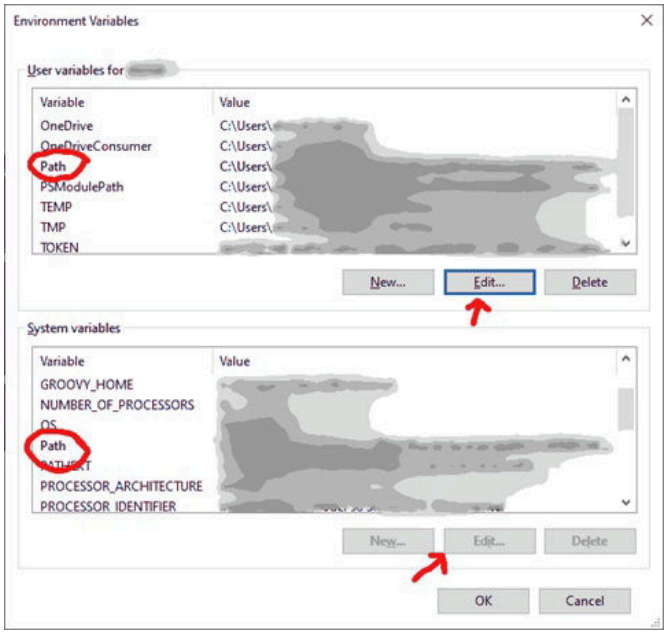
STEP 2: download executable

The executable is available from here:
https://github.com/anacierdem/libdragon-docker/releases/tag/v10.9.1
Once you have the ZIP, you have to place it somewhere where the PATH variable can find it.

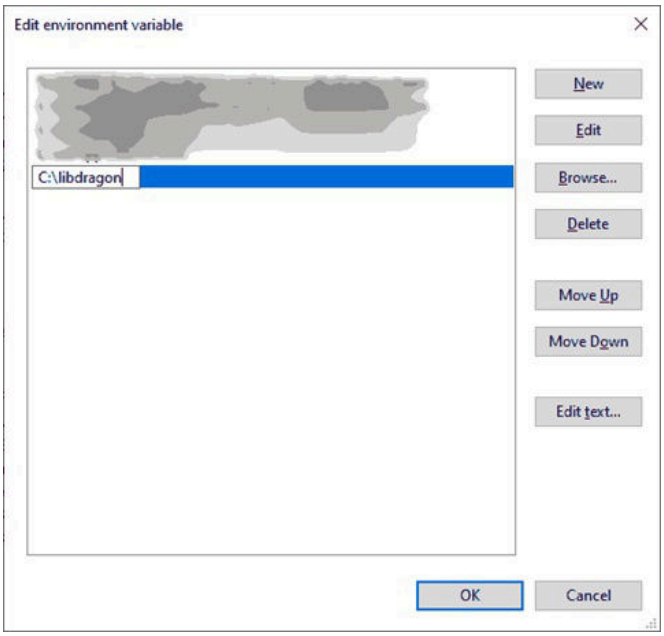
Once secured edit the environment variables like this:



Change the environment variable called "path" or "PATH" for your user.



Add a new entry with the folder where the executable is located



Step 3: Initialize a Libdragon project

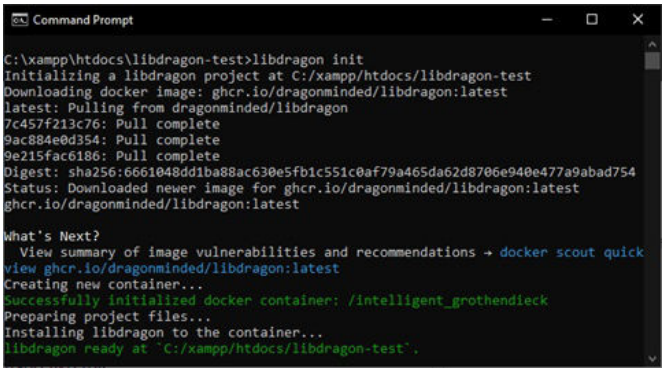
Open your terminal and go to the project folder called using CMD like this:

cd C:\path\of\your\folder

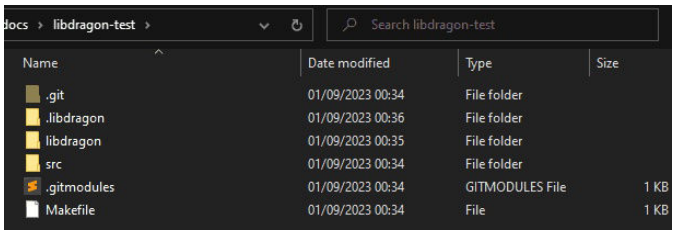
Then run the command to initialize in this folder:

libdragon init

It will take a few minutes to download the necessary files, but you will get something like this:



Now you'll be ready to start developing with Libdragon!

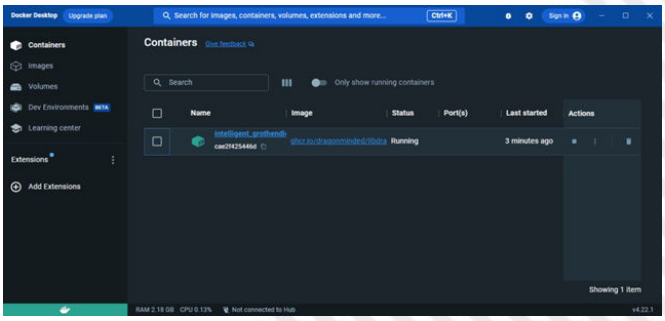


YOUR FIRST PROJECT

At this point you will have finished configuring Libdragon and will have a folder with a GIT repository that you can use for version control. There are several guides on how to use GIT so I won't go into detail here. The main things to know though are these:

- GIT - The folder containing all the data for your repository
- .libdragon - contains the Libdragon configuration
- libdragon - the device containing the submodule. Basically, all the libdragon code.
- src - This is where you enter the code for your project.
- .gitmodules - Contains information about the modules used by the project.
- Makefile - a basic Makefile that contains instructions for building your ROM.

Also, if you look at the Docker desktop window, you will see that a Libdragon image is set up. The name of the image is randomly generated.



The next article will discuss the Libdragon Console. We will take a look at how the console can be modified, manipulated, and how to develop and debug.

A chat with the Knights of Bytes

by Carlo Nithaiah Del Mar Pirazzini

Guys, thank you for this interview and thank you for your great games (Sam's Journey Ed).

Let's start with the first question: how Knights of Bytes came about.

Chester Kollschen: I have been creating video games since the mid-1980s, when I bought my first home computer and started doing it with programming. It started as a hobby with simple text adventures and basic action games, but quickly became a passion as the projects got bigger, involved more people, and paid off by turning out significantly better.

In the mid-1990s I decided to turn my passion into a profession and founded Knights of Bytes. Although the 16-bit era was already in full swing, I targeted the universe of the Commodore 64, a computer I grew up with. My first game Ice Guys came out in 1995 and caught the attention of aspiring retro publisher Protovision. In collaboration with them, I also created and published Bomb Mania and then Metal Dust.

In 2005, Knights of Bytes suffered a hiatus, when I started developing and publishing games for mobile phones under a different label. However, I resumed it ten years later after observing the public's increased interest in retrogaming. I asked Stefan Gutsch, whom I met during my time with Protovision where we first worked together, to join my project team. It turned out to be a very fruitful collaboration because we had a very similar project, and I was always looking for a pixel artist and he was always looking for a programmer.
Alex Ney came on board in 2015 when we were designing a huge platform game for the C64 and needed a multi-style audio wizard who was ready for that task.

That huge platform game became known as Sam's Journey. When we released it in 2017 It received extremely high praise from both critics and the public. This success inspired us to do so we ported the game to another 8-bit platform and eventually decided on the NES.

Can you tell us about little Sam? How does the story of this chubby little hero develop and how has he developed over the years.
Chester Kollschen: I'm a big fan of Super Mario Bros. 3 (NES), Kirby's Adventure (NES), and Donkey Kong Series Country (SNES), and I've always wondered why such games never made it to the C64.

In the early 1990s, I decided to give it a try and designed a small sprite character that could change him from looking cute to tough. But my attempts to create a complete game like that failed because of my lack of skills at that time. And so, the disk with that sprite character went back to its box to wait for better times,

Fast forward to 2015... When Stefan and I realized that he also wanted to make a platform game for C64, we started our research during which I came across that very record. We both liked that character and immediately agreed that this would be our hero. We also hit on the idea of giving him the power to transform into different alter egos. After a series of brainstorming sessions, we compiled a list of features we wanted to have in our game and this time, fortunately, we were able to bring him to life.

Our Sam has won many fans in the C64 community, and we are quite sure that this has given him the confidence to travel to other platforms as well.

A photo from the glorious 1990s with Stefan Gutsch (right), game designer and pixel artist

Chester Kollschen in his design studio

Any future projects coming up?

Chester Kollschen: There are some ideas and concepts that keep coming back to us whenever we talk about upcoming projects. Still, it's a pretty colorful bouquet, including another episode of Sam's Journey, a racing game, a gritty action-adventure, and a shooter.

To decide on a candidate, we are currently creating models and prototypes to see what is interesting and what is not. Of course, we will keep you updated on our Knights of Bytes website and social media channels.

Thank you for your responses, we look forward to seeing more of your titles.

The site where you can follow, purchase and download Knights of Bytes products:
- https://www.knightsofbytes.games/

Alex Ney, Composer and Chiptune Artist

An in-depth look at Amiga38

Article originally published in blog.amigaguru.com

by Gianluca Girelli

Long live the Amiga, the computer that cannot be killed, or... the one that refuses to die!

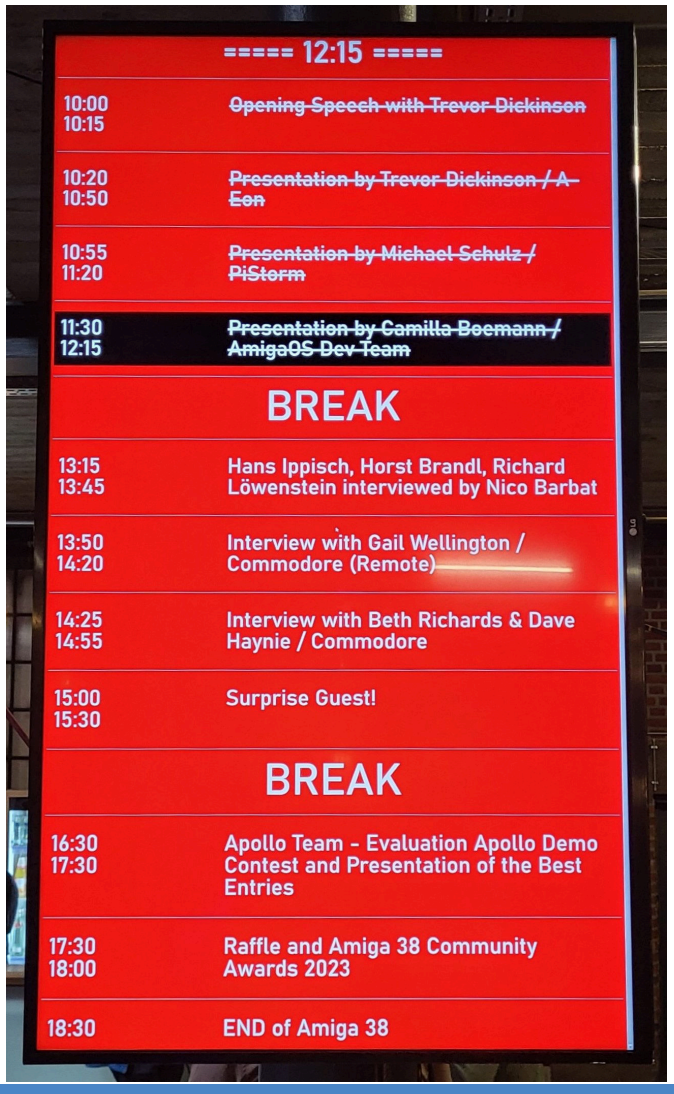
How many times have you heard these words? How many times have you wondered if the concept behind these words was real or simply illusory? How many times have you wished to be, at least once in your life, part of the show, but never had the chance? All these questions, and many more, invaded my mind for years until I had the chance to attend an Amiga event. The year was 2019, the place was Neuss (Germany) and the event was the 34th anniversary of the first Amiga presentation in 1985.

My long-time readers know how I felt for years, due to the fact that life forced me to miss important events in Commodore and Amiga history. They also know that that 'gap' in my soul was finally filled after the 2019 reunion. After Amiga34, I had promised myself to attend these meetings regularly, but COVID and personal life issues (again) decided otherwise. Having skipped Amiga37, I was so excited when, as a surprise, the present edition was announced; I managed to reconcile life and work, booked my flight, bought my ticket to the event, packed my bags and off I went. That's how it went this time.

Organising such an event is no easy task (thank you, Markus!), especially at short notice, so this year the fair took place on just one day instead of the usual two, and exactly as in the previous edition the chosen location was the Kunstwerk in Monchengladbach-Wickrath. Compared to the one in Neuss, the complex is much larger and can accommodate a larger number of visitors, with almost 800 in attendance. Reaching the place is not difficult, thanks to the network of trains and buses connecting the nearby cities. Coming from Italy, I personally booked a flight to Dusseldorf (via Munich) and decided to rent a car, a task that was fulfilled by Enrico Vidale (A-Cube Systems), who arrived in Germany via Cologne and picked us up at Dusseldorf airport. With him he brought a new batch of Sam460LE motherboards along with a now populated PPC laptop motherboard and its case. The project still has some way to go before it reaches its destination, but nevertheless, after years of waiting, we now have something very tangible to contend with.

Although the event was scheduled to start on Saturday the 7th, the party had already begun in the early hours of Friday morning, when Amiga fans from all over the world began to gather at the airport. Despite the fact that 4 years had already passed, legends like Markus Tillmann (head organiser of the event), Dave Haynie (former Commodore engineer) and Mike Battilana (Cloanto) greeted me like a long-time friend and as if we had only left each other a few days before.

After a quick lunch, we went to the Kunstwerk to set up our stands. In the pictures below, you can get an idea of what it looked like. The first day (Friday 6) was mainly dedicated to preparing the stands and meeting the exhibitors, although visitors were allowed in for the latter part of the day. The evening ended with the launch party



and the presentation of the exceptional 'Reshoot Proxima 3' game by Richard Lowenstein and his team, with music by Martin Ashman and Siegfried Karcher, and speeches by Andre Kudra on demoscene. The atmosphere was already at its best, although officially the fair had not yet started!

Saturday arrived in all its glory, with visitors queuing outside the convention centre from the early hours of the morning. Only one YEAR had passed and exhibitors and visitors alike were eager to know how much progress the community had made in such a short space of time. The opening speech was given by Trevor Dickinson, who also took stock of A-Eon's projects. In particular, he shared some details on how the realisation of the Tabor A1222+ motherboard finally came to fruition, thanks to an exciting collaboration between Amiga technology companies: ACube Systems, AAA Technology and A-EON Technology itself. Amigas equipped with this board were finally fully operational at the A-Eon stand.

Then it was the turn of Michael Schultz (PiStorm), soon followed by Camilla Boemann. Camilla is part of the AmigaOS Dev Team and gave interesting information about bug fixes for the current version of AmigaOS for classics (v3.2.2) and improvements planned for v3.2.3. He also stated that AOS3.3 is partly in the making (first alpha stage and only for the classics version) and will arrive no sooner than 2-4 years from now. For the time being, therefore, enjoy what you have: it is already a dream!

The afternoon was devoted to a series of interviews: Hans Ippisch, Horst Brabdl, Richard Lowenstein, Gail Wellington, Beth Richards and Dave Haynie took turns on stage and, after a presentation by the Apollo team, the event concluded with the traditional raffle and prize-giving.

Meanwhile, this time also wearing the role of Retro

Magazine editor, I had the chance to visit almost every single stand and chat with exhibitors, Commodore/Amiga legends and/or long-time friends. These were: Dave Haynie, RJ Mical and David Pleasance (no introduction necessary, LOL!), Trevor Dickinson and Matthew Leamann (A-Eon Technologies), my friends Enrico Vidale and Philippe Ferrucci (A-Cube Systems), HunoPPC (game developer), Sebastian Eggermont (Commodore Users Europe), Oliver Graf (MEGA65 developer), Camilla Boemann (AmigaOS Dev Team), Michele 'Mike' Battilana (Cloanto), Daniel Mussener (GoldenCode. de), Davide Palombo (with me, co-founder of GDG-Entertain), Kymon and his team of developers, my son Simone. I apologise to all the others with whom I did not get a chance to interact. It was not intentional and I promise to do much better next time.ilAs usual, there were too many noteworthy things, so I will only mention a couple: the third book in David and Trevor's trilogy on the history of Commodore/Amiga (From Vultures To Vampires) and an open-source implementation of a Commodore 128 revision 9 motherboard by Johan Grip.

This machine was my first computer and is still my favourite platform, so I am sure you will forgive me for this honourable mention. The project website contains a lot of information about the board's construction, the most noteworthy of which is, in my humble opinion, the integration of a replacement RF modulator.

This modulator fits entirely into the footprint of the original one. If you wish to use an original RF modulator or another replacement board, you can leave these components unpopulated, add a layer of insulation and install an original modulator.

Among other things, there was a fun cosplay of Ravenlordess and her friend. Cosplays usually don't feature much at computer fairs, but this one was special, taking a well-known scene from the legendary 'Monkey Island'.

This time I didn't have the chance to attend the after party (private party for exhibitors only held at the end of the kermesse), but the ways of the Amiga are endless and so we found ourselves (well, at least some of us), the next day, reunited under the Amiga flag as we waited to board the planes that would take us home.

Once again, I cannot thank Markus and his staff (Torsten Raudssus, Paul Kitching and Vitus Zeel) enough for organising such a show and making us feel at home. Their love and passion are remarkable and it shows. Now,

despite the sadness for a weekend that seemed too short, I look forward to the next big event: the 40th anniversary of the Amiga! Stay tuned.

References

https://www.amigashop.org/product_info.php?products_id=391
http://a-eon.biz/
https://www.facebook.com/apolloaccelerators/
https://c128.se/
https://twitter.com/Ravenlordess

Japan, part 23:

Nintendo & Microsoft: once upon a time there were simple drones...

by Michele Ugolini

Dear readers, what retrogaming news is coming from our beloved Japan?
What is happening in the big game creation houses?
Is everything standing still or is everything secretly in motion? Is the dust settling on the great honors of the past, or are there great inventions and innovations about to hit the market? Are their big game companies recycling important titles from the past, or are they creating new strategic parterships to make improvements on some great recent titles?
Here we unveil some news from the past, evolved into an uncertain present, with great insights for the future-a circular philosophy typical of the Japanese land, as I have frequently mentioned in past articles!

Let's make a brief stop at the Big "N" house.
As we know Gamescom is over and there is great buzz about the hardware mounted in the Nintendo Switch 2, a console still not officially announced by Kyoto but set to be the long-awaited heir. We probably won't see it under the Christmas tree, but time is running out, perhaps we will see it in mid-2024, and its performance will be similar to the Xbox Series S and ASUS ROG Ally in docked and portable modes, respectively.
The console's support for NVIDIA's DLSS appears to be confirmed in a job ad posted by Nintendo itself. Aimed at an aspiring data engineer, the open position is for experts in the field of machine learning who are eager to put the technology to the test on "integrated low-power platforms."
The future Nintendo engineer, the announcement says, will be tasked with exploring "machine learning optimization" processes in order to adapt them to "limits in terms of memory, power and CPU." So apparently the architecture of this heir seems to have been confirmed by the big bosses of the big "N."
In the meantime that Super Mario Bros Wonder will come to light we also have news about the Xbox Game Pass: it will undergo a price increase given the planet-wide inflation.

How is it that this brief stop in the big "N" house is pointing its gaze toward the Microsoft house?
Phil Spencer as we know is the head of Microsoft's Gaming division, and in addition to recently declaring such a future increase, as well as reiterating the crisis of AAA-based games, he also announced a surprising thing, an earthquake with an ancient flavor and future visions.
Although nothing can be officially announced as yet, Xbox Game Studios is at work on several titles in collaboration with Japanese studios. We imagine

then that these are titles developed in the Xbox Game Studios Publishing section, which is responsible for producing First Party titles made by third-party developers. No titles have been announced, release predictions are therefore at an unspecified date as Microsoft probably wants to avoid making the same mistake it made with Scalebound, fueling so many rumors only to have them burst like a big bubble.

Not only that.
Microsoft's willingness to expand through acquisitions of major studios is now well known news to all.

America and Japan want to make more and more agreements. This vision of partership should be seen not only from the point of view of video games but especially from the point of view of sharing knowledge of the highest level electronics, the top secret kind, with a veiled military aftertaste, always remaining at the top of productions given the sudden rise of other competitors such as China inherent to any field, worldwide.
In addition, the pooling of knowledge regarding electronics would bring a general upgrade of all the small- to medium-sized industries that revolve around the entire process of CPU/GPU development- from simple chemistry for PCB printing to the generation of new patents given the current stagnation of a Taiwan that focuses investment more and more on its own defense instead of developing new, higher-performance components.
Hopefully, the sovereignist ambitions of some states will not further undermine world stability. Hopefully, we will soon be able to speak only of progress and prosperity for all and perhaps remember the word "drone" solely as a toy equipped with a remote control that allowed us to observe a leather balloon stuck on the roof of our house.

Back to Microsoft... In a recent e-mail exchange that came to light as part of the proposed Activision-Blizzard acquisition,
Phil Spencer reportedly wrote to a colleague, saying that the acquisition of Nintendo would represent a "career-defining moment" for him. So it appears that Spencer has been courting Nintendo, but that the big "N" has (apparently as well as momentarily) declined to be bought by the American giant.
Here is a translation of what was written in the e-mail that emerged :
"Takeshi, I totally agree that Nintendo is THE primary ASSET for Gaming and as of today Gaming is the most likely path to beat for relevance to consumers. I have had numerous conversations with the team leading Nintendo about closer collaboration and feel that if a company in the United States had any chance with Nintendo, we are probably in the best position. The unfortunate situation (or fortunate for Nintendo) is that Nintendo is sitting on a mountain of cash, and they have a board of directors that has not pushed for further increases in market growth or increases in the value of their shares, except in more recent times. I say "in more recent times" as our ex-Value Act on the Microsoft board acquired heavy shares in Nintendo, and I kept in touch with Mason Morfit as he proceeded with this acquisition.

It is likely to continue to push for more Nintendo shares, which could create opportunities for us. Without this catalyst, I don't see a chance for a short-term acquisition of Nintendo where both parties agree, and Microsoft and I don't believe that aggressive action would be the indicated one, so we are playing the long game. However, our board has viewed the entire written report on Nintendo (and Valve) and is totally supportive on both if the opportunity presents itself, as am I.

I like this discussion and appreciate that you see opportunities in it. At some point, acquiring Nintendo would be a career-defining moment, and I firmly believe it would be a great move for both companies. Nintendo is just taking so long to see that their future exists outside of their hardware."

Let us conclude here for now guys, as you can well see in Japan the past, the present and especially the future always coincide as a spiral movement. Never underestimate even a bend of this spiral: it could hide great surprises accrued from experience in not making the same mistakes of one's past again.

Until the next reading dear friends!

GP Cars – Commodore 64

by Mic the Biker Novarina

Release Date: March 19, 2023

Released by: LHS

This game was released in March 2023 by LHS. The team in charge of the technical side of GP Cars is one of the highest quality: LSH produced the code; Almighty God designed the graphics, here assisted by Nico Galis; and NecroPolo handled the sound.

Before its official release, the title already made headlines, as it was ranked 2nd in the WiLD Demo Competition FOReEVER 2023 - ROBOT. It is an overhead-view driving game, a genre that was definitely in fashion in the golden age. Just remember timeless masterpieces such as Micro Machines and Supercars on the cookie, and Indy Heat on the Amiga. But from the very first graphic screens of presentation I gasped, because I had a flashback called "Hot Rod," a Sega-branded racing game that not everyone will remember. As I began to crank out the first few games, I got confirmation that this GP Cars draws heavily from the Sega arcade, released in 1988 in arcades, and converted in 1990 for major domestic platforms.

We are lucky after all, because today we have access to remarkable games, like this one, at no cost. A game that at the time would have cost several thousand liras, or a tenner in the pirated version. We do Load and shortly afterwards we are greeted by the chugging of the engine effectively reproduced by the sound waves of the SID. The screen gives us beautiful colorful graphics, with a blue muscle car being chased by what looks like a red Porsche.

Here we come to the main screen, where we will see the five locations in which we can engage. Each location has five tracks, for a total of no less than 25 races available. Not so easy, because you have to win the races in the first area to gain access to the next one and so on. We start with the first slot of races, and are greeted by a 3,2,1 countdown, GO with a good digitized vowel. Races can be against the stopwatch or against other vehicles, with a total of four. Initially, the madonnas fly like drones in the sky, as the borders marking the road are like concrete walls and slow us down at the slightest contact. The track is not wide at all and the curves are often tight. Translated in a nutshell, the first ten minutes I spent pounding all over the place before realizing that in the turns it is wise to adjust speed. Having finished one lap decently the other scenarios begin to unlock, in which one often finds oneself richly sporting with opponents. I remain displaced by the nonexistent sprite collision: the cars can pass over each other without a problem. Probably if they had combined the difficulty of not hitting the limits of the road with the collision between sprites the game would have been unplayable.

CPU-driven cars literally walk away initially, but as the games go on, the feeling improves and I play it turn by turn. It still takes a while to master the inertia of the controls, however, with the car increasing steering speed while holding direction. Particular attention must be paid

to the construction sites scattered around the roads: bumping into one will heavily damage our four-wheeled vehicle, much faster than just hitting killer guard rails. Here and there you can pick up some valuable Nitro that gives us that extra bit of grit. As we progress through the game we can choose different types of cars and change their color. This is purely an aesthetic frill, because the performance among the various little cars in the game is always the same.

Graphics and Sound

The graphics department rocks, no doubt about it. The Commodore 64 palette is put to good use, and the color choices are always great. We also have some animation in the background, a real treat for the eyes. The sprites are detailed and well colored: each car type is well recognizable and characterized ad hoc. Scrolling is top notch, smooth in every direction, accurate and fast.

The audio compartment is TOP SID. Starting with the effects, as mentioned at the beginning of the article, and ending with the multiple in-game songs, the audio is a sight to behold. The songs give just the right amount of charge, are well composed, and "sound good." The in-game sound effects are just where they are needed: skidding and bumping. In this way, resources and channels of the SID are not wasted noticeably.

Biker's Thoughts

I was very intrigued by this GP Cars, and as soon as I had a few minutes of time I downloaded it and started playing it. The merits are many, despite the not inconsiderable difficulty. Graphics and sound get us excited, and once you get the hang of the dynamics of the controls you have a great time. As a fan of this genre, however, I have some regrets: with little more this GP Cars would have been phenomenal. For example, I would have liked to have chosen a driver at the beginning and maybe seen a podium with an award ceremony at the end of the races. A workshop area where the car could be repaired and perhaps upgraded in road holding and performance would have been a spectacle.

Of course, these are all things we've seen before in other games, but precisely because I've played them all I notice that details like these would have taken it to another level. I loved, and still love, good old arcades with top or iso-top views, and GP Cars for me, to put it "Zzap!" style, is definitely a "hot game."

OUR FINAL SCORE

GAMEPLAY: 85% Initially hostile, once you get the hang of it you start having serious fun!

LONGEVITY: 80% There is no shortage of challenges, no shortage of tracks either so the challenge is assured. Likely if you're not really into the genre the "I can't save" frustration may come, but when playing Super Sprint was it or wasn't it the same thing?

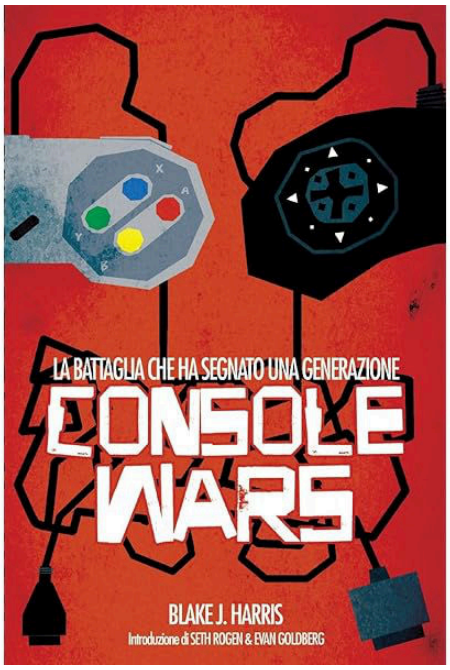
RMW Recommends: Videogame Readings

by Giuseppe Rinella

I don't know about you, but with the arrival of autumn and, even more so, the following winter, I spend most of my free time indoors. In short, the weather outside certainly does not invite spending hours outdoors.
Among the activities I love to do at home is of course the abuse of retro and non-retro video games (family permitting, which always comes first, it goes without saying), also I become a devourer of books so very often I put aside the joystick or joypad and immerse myself in the written pages.
I have always been extremely interested in everything related to the history of our favorite pastime, and so I am always on the lookout for books on the subject.
Therefore, I thought I would suggest a series of books in my humble opinion that you absolutely must read, as we will see in some cases these are really wonderful items that you will find yourself flipping through from time to time, perhaps with a clear eye.
Small clarification: we do not advertise here to authors or publishers, we do not receive any sponsorship, so if you notice some enthusiasm on my part it is not to entice you to purchase, it is simple joy!

Let's go then!

CONSOLE WARS (ITA-ENG)
The first real, epic "console war" was, as we all know, the one that saw Sega and Nintendo clash with the arrival of the 16-bit console generation. Until then the market was basically monopolized by one company, with others chasing without much hope of keeping up. First there was Atari and then, after the 1983 crisis that nearly wiped out the video game industry, Nintendo, which for years was synonymous with home video games with its wonderful FamiCom/NES.
In 1990 Hayao Nakayama, then president of Sega, decided to openly challenge Nintendo, which dominated the 8-bit era. To do so, he physically went and picked up Tom Kalinske on a family vacation to propose that he become CEO of Sega's American division, to accomplish what then seemed an impossible mission. Tom Kalinske boasted a distinguished career at Mattel and until then had never had anything to do with video games.



Kalinske agreed; the Megadrive was already on the market in Japan and would shortly be coming to the U.S. and Europe. He was responsible for aggressive marketing campaigns, to say the least (famous was the slogan "Genesis does what Nintendon't"), which definitely aimed at a more grown-up audience than Nintendo's target audience. This insight proved to be decidedly spot on. Aggressive pricing to say the least, a mascot like Sonic, and the Megadrive's stock of titles did the rest, and it all led, at least for a period, to Sega playing on equal terms against Nintendo, even surpassing it in terms of sales, although it did not last long.
Console Wars chronicles all this and does it in a great way, enriching it with testimonies and anecdotes, all told at a pace that keeps the reader glued.
One might object that the author, Blake J. Harris, is a bit biased by blatantly sympathizing with Sega, and I totally agree with that position. The purpose of the book is not to determine who is the better man in this battle, but to tell a story that is somewhat reminiscent of the classic David versus Goliath, Sega's almost insane idea to take on the behemoth Nintendo against whom no one could do anything.
When the 16-bit era ended, the victory was undoubtedly Nintendo's from any point of view, the Super Nes proving in fact superior on all fronts against the Megadrive. Sega,

from 1993 onward, began to lose ground, and the launch of questionable or inadequately supported products by Sega itself (Sega Mega CD and shortly thereafter the Sega Saturn) led to Kalinske's abandonment, and Sega began its not-so-slow decline on the home game front, at least as a hardware manufacturer.

Console Wars is a beautiful and exciting book, a lesson in marketing that makes unfairness and aggressiveness its stylistic hallmark, perhaps no longer viable in today's day and age but extremely interesting, definitely retrieve it!

COMMODORE AMIGA: A VISUAL COMPENDIUM (ENG)

If you already don't know who they are, definitely mark "Bitmap Books," a publisher that churns out books of truly sublime quality.

This is not the only book of theirs that I am going to talk about, but as a true worshipper of all things Amiga I couldn't help but start with this one.

The book traces the history of the Amiga by offering a roundup of the most significant titles released for the never-to-be-loved Commodore machine.

Two pages are dedicated to each game, which are filled with a large image depicting it, accompanied by quotes from developers or others regarding the title in question. There are also insights of various kinds: the historic software houses that gave us epic titles, interviews, trivia (why is the left hand used in the historic image of the hand holding the floppy? Where did "Guru meditation" come from?), a substantial section devoted to the demo scene, titles that were never released, and much more.

Each page is a triumph of pixels, flipping through this book for those who, like me, literally grew up with Amiga, is like flipping through an album of old photos that make me remember the emotions those titles were able to give me. The quality of the printing, the binding, the paper used, everything is really of the highest standard, the book is made by enthusiasts for enthusiasts, and the care and love reposed is clearly visible on every page.

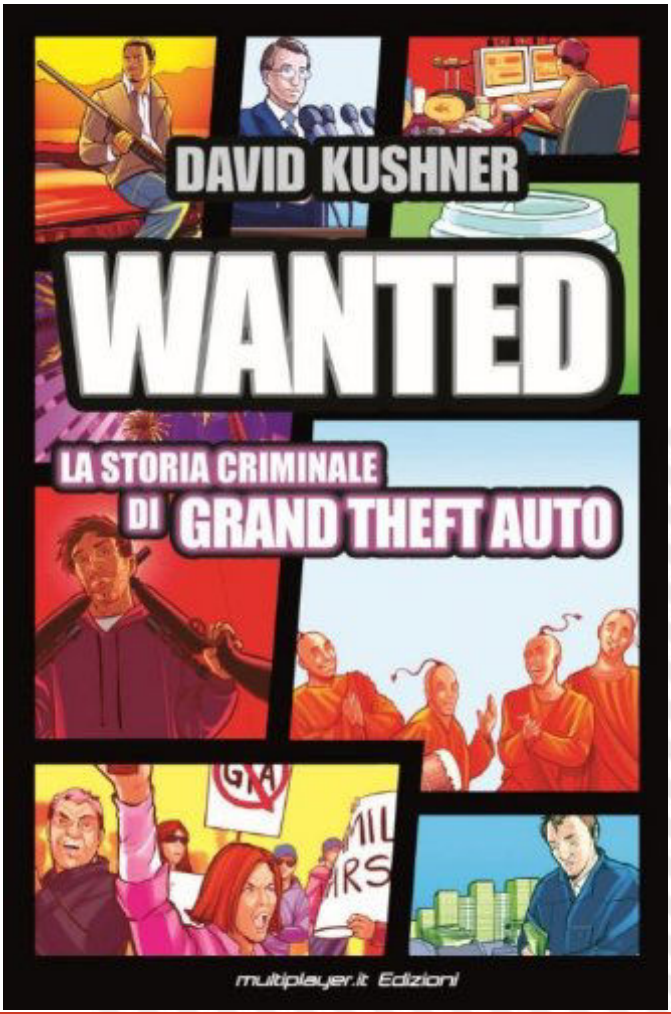
I could make just two notes (dictated by pure personal taste): first, for some games perhaps the most significant images were not chosen. Second, Shadow Fighter is missing among the titles, sacrilege! No big deal though. There are other books in the same series dedicated to other platforms: Commodore 64, Sega Master System, ZX Spectrum, NES, SNES, and Atari 2600/7800. Whatever your favorites are (slowly, keeping an eye on the family budget, I will get them all!), buy them with your eyes closed, you will then have in your hands a product of sublime quality that you will find yourself flipping through over and over again, guaranteed.

WANTED: The Criminal History of Grand Theft Auto (ITA-ENG)

If like me you fully experienced the birth of the Grand Theft Auto phenomenon back in 1997 and have not missed a chapter of the saga from there on, this book is definitely for you.

Wanted chronicles the birth of GTA and the subsequent overwhelming growth of Rockstar Games, starting with the first episode and ending with the launch of GTA IV.

Thanks to the testimonies of those who lived through



those years firsthand and a great deal of research work by author David Kushner (whom we will meet again later), it is possible to relive the crazy story of a series that marked a before and after in the history of video games, shattering every sales record and becoming a true phenomenon of custom.

Reading this volume it becomes clear how the name "Rockstar" is by no means accidental, the studio founded by Sam Houser was chock full of genius and even more unruliness. Frantic work rhythms, advertising campaigns that were controversial to say the least, battles against politicians and parents' associations that tried in every way to hinder the irresistible success of GTA, all fictionalized just enough to avoid falling into the cold chronicle of events, which would risk boring.

In short, there is everything to keep the reader glued, including the account of the famous "hot coffee" scandal. Wanted is a truly enjoyable and exciting read, if you have loved and enjoy the GTA series it is an absolute must, but even otherwise it remains an extremely interesting and compelling read for anyone who wants to delve into a fundamental piece of video game history.

VIDEO GAMES: STAGE-ONE (ITA)

If you have come this far, the topic of video games evidently interests you, and "VIDEO-GAMES" is therefore an absolute must-have volume.

The book traces the history of video games from its beginnings to 1979, but this is only the "STAGE-ONE," the second volume (1980-1984) is already available while the third (1985-1989) will be released later this year. The series will consist of five total volumes, coming to cover the period 1972-1999.

The work done by Andrea Contato, an author whom I encourage you to follow, is truly impressive. The thorough and meticulous research is evident, as is the author's dedication and passion. The book is full of quotations and interviews of the people who contributed to the birth of video games, true visionary geniuses of computer science.

It starts from 1972 as mentioned, the year of PONG's birth and everything that came after, from the first online game to text adventures, from Space Invaders to Asteroids and more, with incredible documentation work.

The final appendix, with cards of home systems that came into being during the period covered in this volume, completes the whole. Pong, Odyssey, Intellivision, Apple II, Commodore PET and others, nothing is really missing.

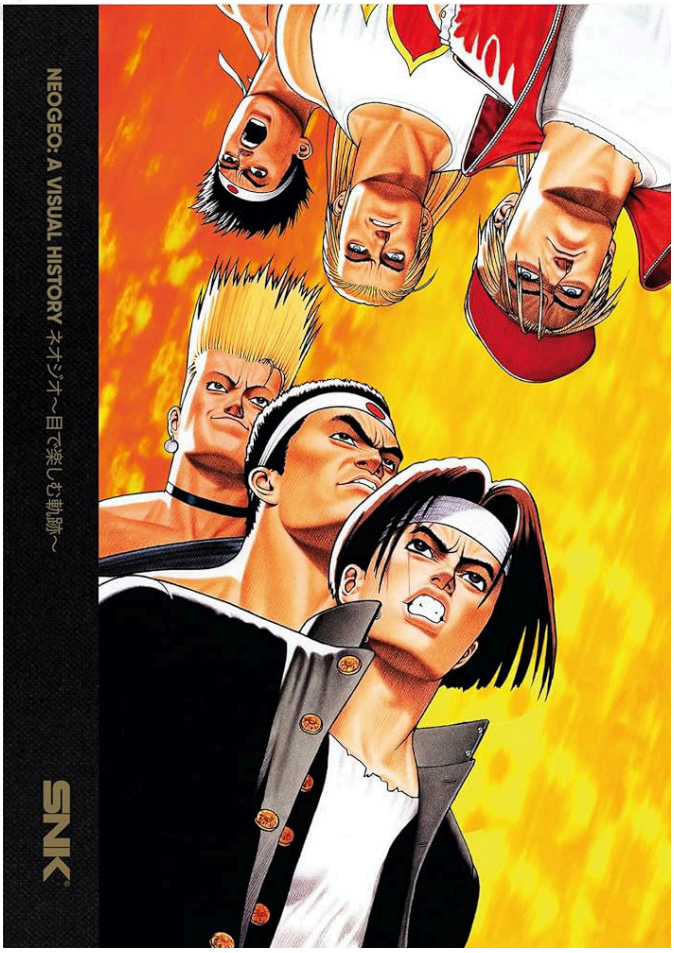
I wrote about "STAGE-ONE" because since it is the first

volume of the series it would be logical to start here, my totally positive judgment is exactly the same for the second volume and, I am sure, for those to come out in the future.

As mentioned a few lines above, VIDEO-GAMES is a book that should not be missing on your shelves, it oozes love and passion for the history of video games from every page, the work of Andrea Contato (with whom I had the pleasure of having a chat on more than one occasion, if you attend fairs and events related to retrogaming and retro-computers you are very likely to meet him with his stall, go see him, you will find an affable and very helpful person) is almost touching, I highly recommend you don't miss it and the same goes for the following volumes, I can't wait for the third one to come out!

NEOGEO: A VISUAL HISTORY (ENG)

Those who, like me, grew up on bread and video games in the 1980s/90s can easily understand how much and why NEOGEO constituted the ultimate possible in video games. In my neck of the woods it was synonymous with highly spectacular games that could only be approached, however, in the arcade. We all knew of the existence of the NEOGEO home console, but it was for us something mythological and unattainable, so much so that we doubted its real existence, while the price lists that appeared in



trade magazines told us that yes, it did exist and it was possible for us to admire it just in pictures.

This beautiful book, also published by Bitmap Books under official SNK license, traces the history of this fantastic system and does so mainly in pictures, as is easily guessed from the title. Not only those though, the history of the system produced by SNK is told accompanied by interviews with developers and collectors and various trivia, but undoubtedly it is the pictures that steal the show and the volume's considerable size (30X22) does it credit.

It starts with a series of beautiful full-page photos depicting all of the hardware produced by the glorious Japanese company; it's all a triumph of booths, consoles, controllers, and more.

This is followed by Box Art and later by sketches and character studies of some of the most iconic games, including Art of Fighting, Fatal Fury, Samurai Showdown, and Metal Slug.

It is the next part that probably sparks the enthusiasms of those who have always admired NEOGEO titles, mine certainly, the one devoted to Pixel Art, which I personally consider an art form and adore (who doesn't?). A succession of huge images that allow you to appreciate every single pixel of every single sprite, scenery and cut scene, this is something bordering on pornography.

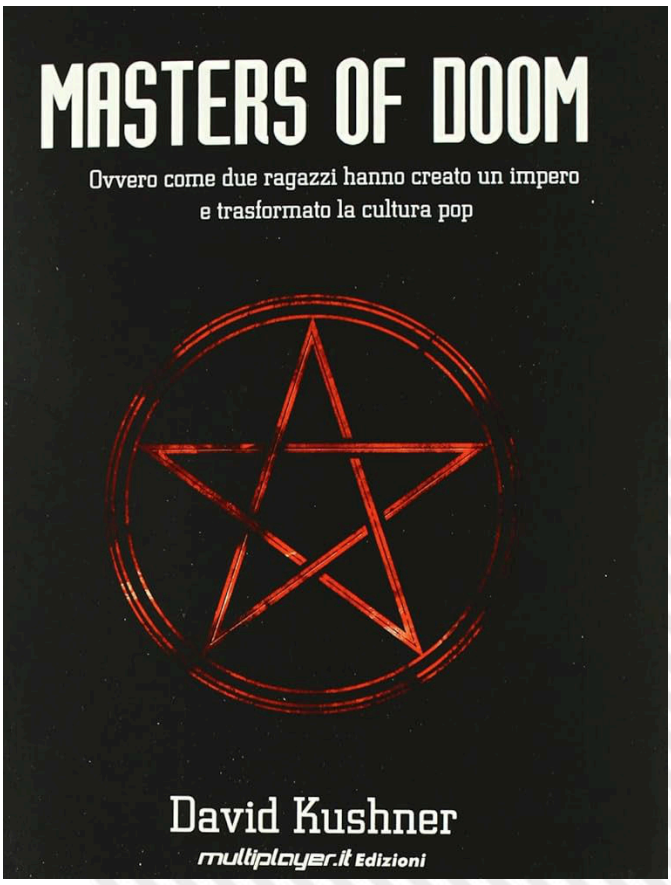
NEOGEO: A VISUAL HISTORY is a great object, and the hardcover and fine paper give it the importance it deserves. It is certainly not the book to take on the subway, the size and weight of about two pounds does not make it particularly comfortable to carry, but trust me, if you have lived in the myth of NEOGEO like yours truly, this fine volume will make you very happy.

MASTERS OF DOOM (ITA-ENG)

We find David Kushner again (this volume predates Wanted actually, so that is where we find him, but it changes little), and here again the author recounts the birth and meteoric growth of one of the software houses that made video game history, ID Software, and its title after which nothing would ever be the same: Doom.

Masters of Doom traces the incredible story of ID's two founders, John Carmack and John Romero, genuine geniuses with profoundly different personalities and perhaps because of that, at least up to a certain point in this story, complementary. So different that they are called, in the opening chapters, "The Rockstar" (Romero) and "The Scientist" (Carmack).

We will read about the exploits of two authentic revolutionary rebels (at least in the video game sphere), starting with the release of Commander Keen (after the conversion of Super Mario Bros 3 they made for PC was rejected by Nintendo) that took advantage of the revolutionary technology called "Adaptive tile refresh" created by



Carmack and that made horizontal scrolling possible on PC, something never seen until then. And then Wolfenstein 3D where it was already possible to catch a glimpse of what was to come next, all the way to Doom that shocked the video game world and beyond, becoming a genuine pop phenomenon, and later Quake.

It is possible to savor, thanks to this volume, the atmosphere of explosive creativity present in ID Software, a group of true genius nerds, and of nights spent creating something unique and never seen before.

The resounding success achieved by Doom and subsequent episodes, the total freedom given to fans who began creating level editors and new versions of Doom (including a Star Wars-themed one) all but hindered the two Johns.

By reading Masters of Doom you will not only learn more about the birth of a true video game milestone, you will also read about something unique and probably unrepeatable in the history of our favorite medium.

A must-read book, entertaining and exciting, excellently written by David Kushner with an overwhelming pace.

Only problem for the Italian edition: finding this book new is impossible, you will have to turn to the second-hand market as I did a few years ago, otherwise there is always the English version that is easier to find.

Absolutely retrieve it if you haven't yet, few books that tell a piece of video game history like this one have excited me as much.

THE ART OF POINT AND CLICK (ENG)

Here comes Bitmap Book again with another great book, great in every sense of the word, again being a volume of significant size and weight that certainly does not make portability its strong point.

Unlike other Bitmap Books books, such as the "Compendium" series or volumes like "NEOGEO: A VISUAL HISTORY," what is explored in depth is not the history of a single machine, brand or series of video games (as is the case, for example, in the book "Metal Slug: The ultimate History," also published by Bitmap Books), "The art of point and click" traces the evolution of a genre, from its inception to the present day, namely that of graphic adventures.

The book traces in chronological order the entire history of the genre from the earliest text adventures, through true point-and-click adventures and on to more recent games.

Just as with the other Bitmap Books books, a great deal of space is given to images; here we find a riot of mind-blowing pixel art that benefits from the large format of this massive volume.

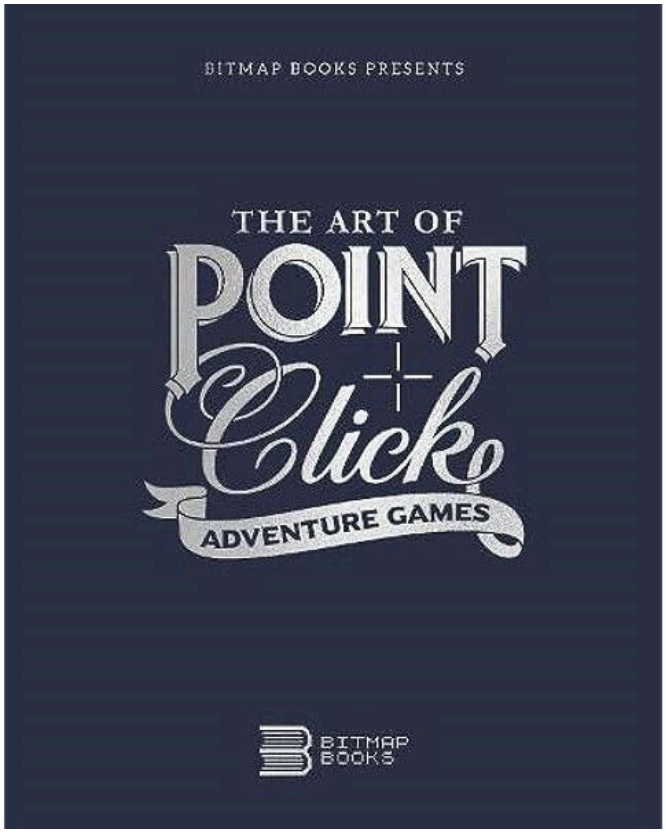
Not only images though, "The art of point and click" also contains more than forty interviews with those who worked on the masterpieces illustrated here, the likes of Al Lowe, Ron Gilbert and Tim Schafer just to name a few of the most famous. The interviews are really interesting and full of trivia, for example, I didn't know that Ridley Scott never gave a damn about the game dedicated to Blade Runner.

"The art of point and click" is an absolutely essential book for those who have always loved this genre of games, but it is extremely interesting for anyone, as well as a joy to behold.

As can be expected from a Bitmap Books product, again the quality of the book as an object is equal to that of the contents, paper and binding are of a high standard, and the hardcover gives it the right elegance.

Small note: the edition in my possession, the first one, goes up to 2017; however, I read that the volume is up to the third edition, which includes 28 additional pages, with new interviews and games not included in previous editions.

An absolute must-have book, even for those like me who have never been very familiar with the genre due to manifest inability (with the exception of the first two Monkey Island, Simon the Sorcerer and little else) but I admit that flipping through this marvel makes me want to dredge up masterpieces and hidden gems that I have



guiltily missed in the past.

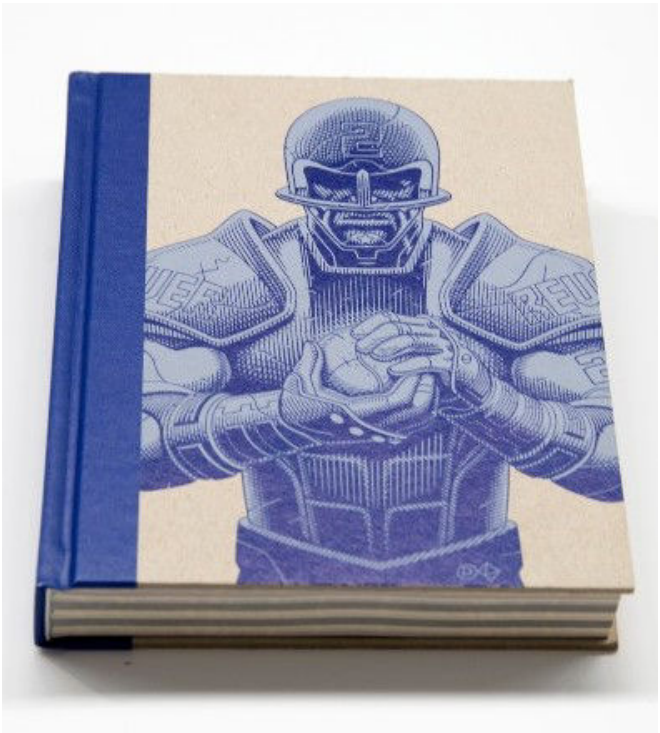
THE BITMAP BROTHERS UNIVERSE (ENG)

Another piece of video game history, because the Bitmap Brothers really made history.

Anyone who lived through the 16-bit era like yours truly, even more so if with an Amiga like yours truly, cannot help but love them, and this fine volume chronicles their heroic exploits.

At a time when developers were mysterious faceless entities and when publishers dictated their own times and conditions, the Bitmap Brothers founded by Steve Kerry, Eric Matthews and Mike Montgomery were a true revolution, putting developers at the center of everything. In their own historic way were the photos, published by industry magazines, in which the three founders were showing off poses and looks that soon earned them the appellation of true "rock stars," something never before seen for mere programmers and that made it clear how much the mission of the British trio was to make sure that those who really made games got the visibility they deserved. Not just form but lots of substance, the games produced by the Bitmaps were always of the highest quality, technically magnificent, with a metallic aesthetic that made them instantly recognizable and exceptional care for the sound aspect.

The Bitmap Brothers Universe traces the history of the glorious software house, later also a publisher with the founding of Renegade, recounting the genesis of each of the titles made.



There is a lot to read but that's not all, sketches, studies of characters (some never used) and environments, a rich gallery of color images of the historic Bitmap Brothers titles (to which a filter simulating CRT TVs has been applied, wonderful) and more, such as sketches of Speedball 3 which unfortunately never saw the light of day.

The quality of the volume is equal to that of the contents, the thick cardboard cover gives the object that massive air that Bitmap Brothers masterpieces conveyed, the choice to use a violet/indigo for the texts instead of the classic black is very apt and perfectly in line with the Bitmap Brothers aesthetic, the "cool" photos mentioned earlier and reproduced here in pixelated monochrome, styled after photos taken with the legendary Gameboy Camera, are a nice touch.

An absolutely recommended book, the publisher Read-Only Memory then is always to keep an eye on, I suggest going to look at their catalog which contains works of great quality.

We have come to the end of this small selection of books that I feel like recommending to you, and there would be more, maybe I will come back to this topic in the future, who knows.

If you are as passionate about the history of video games as you are about playing them, check out these books, and if you would like to suggest other volumes that you think are worth reading, we look forward to seeing you on our social channels!

Greetings to all and Amiga Forever!

NEW GAME

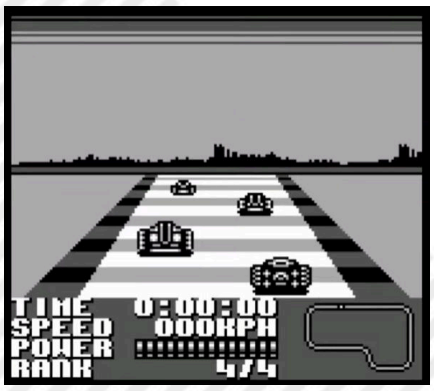
G-ZERO

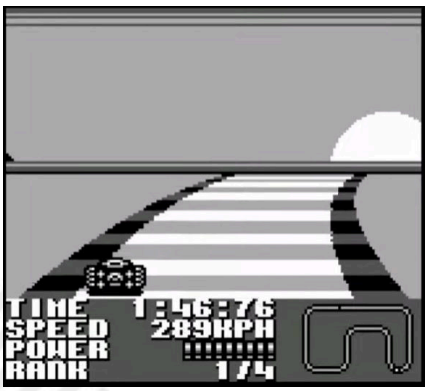
Year: 2022
Editor/Developer: User0X7f
Genre: Driving
Platform: Game Boy
Website: https://user0x7f.itch.io/g-zero

Year 2560. The human race has managed to survive a terrible alien invasion, suffering great loss of life.
To celebrate the big day of victory, a race, the G-ZERO Grand Prix, is held every 4 years! Only the best drivers can participate, and only one will be the winner.

G-ZERO is a simple arcade racing game for Game Boy inspired by the Super Nintendo legend F-ZERO.
In each race one will have to complete two complete laps of the track and try to finish first. To win the trophy you will have to complete all six races present.
Our vehicle (out of three to choose from) has limited power, which decreases every time we touch one of the track edges and an opposing car. If the power runs out, the race will end and it will be a game over.
You use the directional cross to move, the A button accelerates and the B button slows down the car. Finally, the start button pauses during the game and obviously starts the game.
It is a good homebrew, a tribute to its famous big brother born in the early 1990s.
It doesn't have the same sense of speed and gliding mechanics that we loved on Super Nintendo, but there is a good sense of fluidity and good work on the maps present.
The game was developed in C using GBDK, and only the ISR scanline routine is written in ASM.
It can be played in emulation and also on real hardware, which I recommend because it makes the experience much more fulfilling.
It is a small title that is completed after some practice but allows the player to spend a few hours whizzing around the futuristic roads of the future.
Kudos to the developer for this type of product, perhaps not the best of the best but certainly fun and well done.

by **Roberto Del Mar Pirazzini**





OUR FINAL SCORE

» Gameplay 80%
Smooth and with well-structured paths. Perfect controls.

» Longevity 70%
It is neither difficult nor very long, but it is enjoyable to play.

NEW GAME

ATTACK OF PETSCII ROBOTS

Year: 2023
Editor/Developer: The 8-bit-guy/ Robosoft
Genre: Strategy/Adventure
Platform: MSX V9990 con OPL4

Born on Commodore PET and then ported to a host of different systems, 8 Bit Guy's title arrives, thanks to the golden little hands of Robosoft also on our beloved MSX V9990.

The aim of the game is always to try to infiltrate inside some settlements and destroy all the rebel robots. The trick is to find useful tools and use ingenuity instead of simple brute force.

Dynamic and very well thought-out action and excellent gameplay that does not lose interest and keeps the game tension high.

This porting comes out during MSXDev 2023 and was made possible thanks to Aoineko's fantastic MSXgl library and Grauw's Replay libreia for music.

Technically it closely resembles the Sega Megadrive version but with greater fluidity. I found the graphics rich and good-looking.

The title takes advantage of OPL4 FM audio, and I must say it is a great feeling. In emulation this is somewhat lost. On BlueMSX this possibility is excluded and so I recommend, for those who do not have the original hardware, to use OpenMSX for the perfect sound rendering.

The game controls are customizable, but I personally prefer the default keyboard settings over the pad.

As with previous versions, it must be said that it is not an easy game to approach and requires logic and a lot of practice, but it is undoubtedly challenging and enjoyable.

In short, another little gem on MSX. Enjoy it!

by Giampaolo Moraschi

Website: https://www.msxdev.org/2023/09/19/msxdev23-18-attack-of-the-petscii-robots/

OUR FINAL SCORE

» Gameplay 85%
A well-developed and proven title due to numerous versions on many platforms that features good gameplay and well-structured controls.

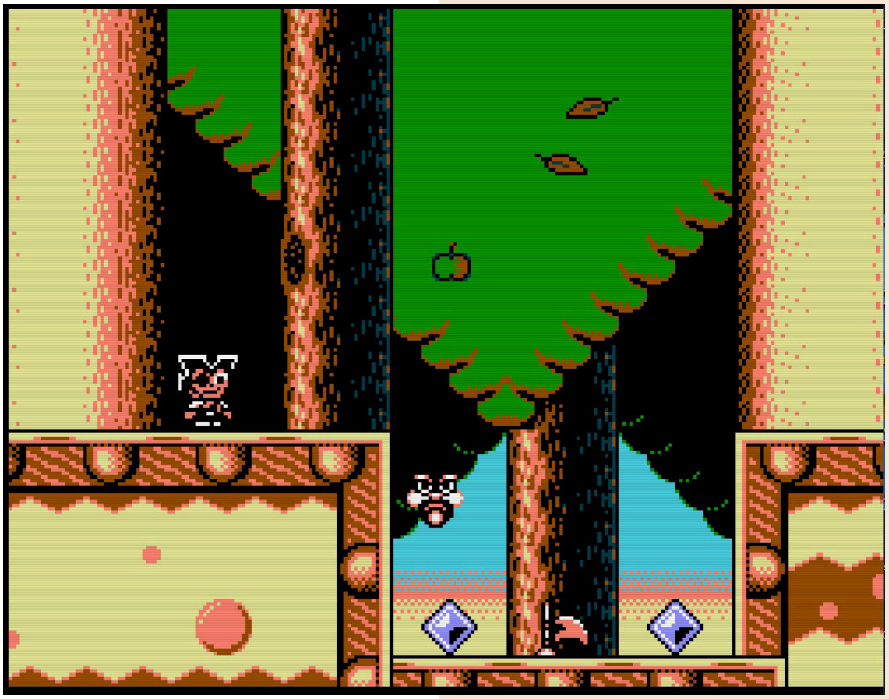
» Longevity 80%
Not for everyone and not for those looking for fast-paced action. But it is solid and enjoyable.

NEW GAME

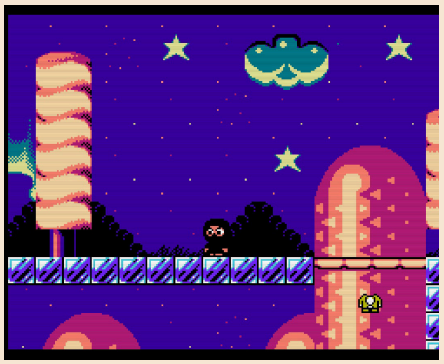
SAM'S JOURNEY

Year: 2023
Editor/Developer: Poly.play, Knights of Bites
Genre: Platform
Platform: Nintendo NES
Website: https://www.knightsofbytes.games/samsjourney/nes

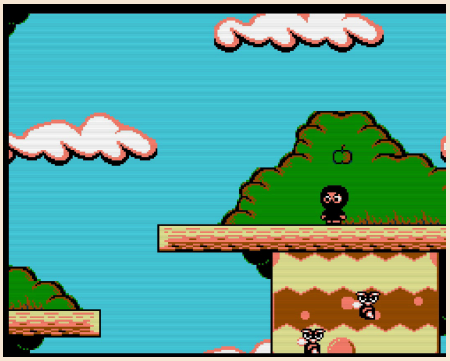
To call Sam's Journey a good game would be an understatement. The overall presentation and gameplay are nothing short of wonderful. Sam's Journey on Commodore 64 proved to be a breath of fresh air and a boost engine for the "new wave" of next-generation titles for our beloved "breadbin."
Smooth, fast directional scrolling that overwhelms the visual senses and a masterful audio compartment and vivid, colorful game environments.
It was so on Commodore 64, and this version on NES is even more glitzy, big and colorful.



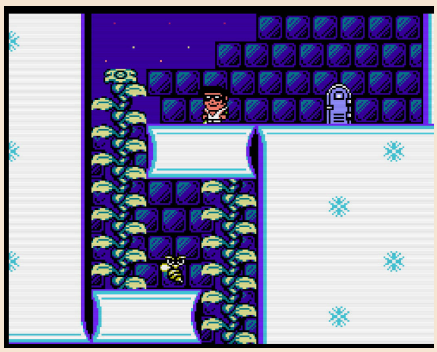
The game remains true to itself. Sam is kidnapped in the middle of the night by a large claw that comes out of a closet. He finds himself in a crazy colorful world that he must explore through 30 levels divided into 3 mega-maps to find his way home.
Game levels littered with a wide range of enemies who want to stop funny little Sam from continuing his journey through the worlds. Our chubby hero will have to use switches, trampolines, secret passageways, hopping owls and collect keys to progress through the various levels and figure out how to use all these objects in the right place or the right way. Special costumes (hidden inside chests or in other locations) will change Sam's appearance and increase his special abilities. Pirate Sam can fight with a saber, Ninja can cling to walls, Disco Sam can twirl around, and so on (there are also Astronaut Sam and Vampire Sam, which are incredible). The costumes are also useful in that they provide a layer of protection from enemies, somewhat like Sir Arthur's armor in Ghost 'n' Goblins, so as to avoid a death from a lethal blow. However, only one costume can be worn at a time, so you must choose whether to change your costume each time you encounter one. Throughout, the game also contains diamonds to collect, coins and collectible trophies that increase the final score.

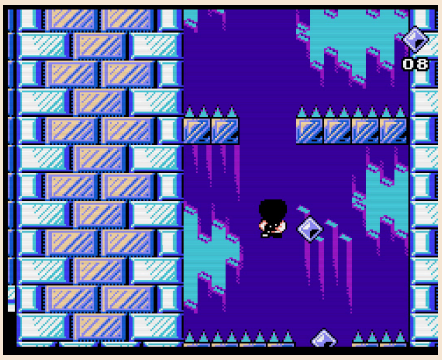


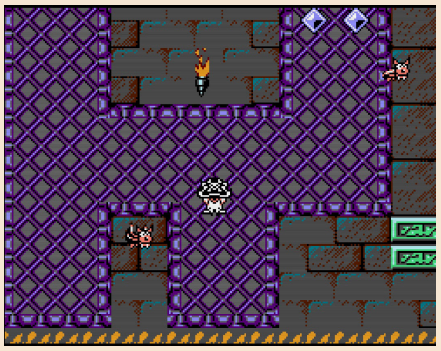
Moving Sam around in his new world on NES is a pleasure. The character, as in his incarnation on Commodore 64, is quite versatile with his ability to run, jump, climb, swim and pick up large objects to throw at enemies.
Enemies are often hostile and all well-defined with different movements and behaviors, which means that eliminating them will often require different approaches.



The gameplay is solid. The levels are well constructed, and the ability to



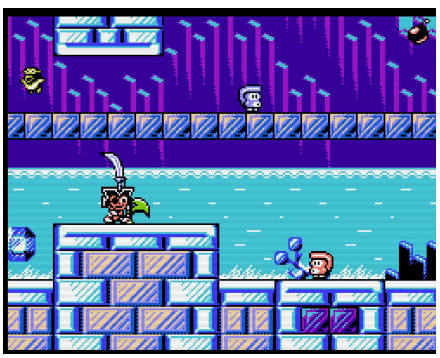




change costumes is a winning and clever feature that introduces strategic elements to approach a battle in a certain way versus another.

The title allows us to save through convenient check-points as was the case in its Commodore 64 counterpart, a really welcome option given the 30 hours of gameplay we will be faced with to tackle the adventure. These check-points allow us to restart from that point in case of death, and this is much appreciated.

We come to the comparisons--is this NES version superior to the Commodore 64 version?
Question we can immediately solve this way--this version has graphics with higher resolution and many more colors. The pad with two buttons allows us to optimize the game better than the single button on C64. The game view is definitely wider and, as I told you, the ability to take advantage of more colors helps a lot.



Audio-wise, I prefer the C64 version. Don't get me wrong, the sound and accompaniment on NES are great, but on C64 the SID has more "flow" and renders better.
As with the previous version, here too I would have preferred the option of selecting the costume instead of automatically applying it, but it is a tiny little hair in the perfect egg that is Sam's Journey.

The Knights of Bytes showed they had it going on again this time, creating one of the finest platformers on the Nintendo NES of the last generation and perhaps one of the finest titles in the genre for this console ever, as they had undoubtedly done by making this game on Commodore.
The title is distributed by Poly.play in cartridge format and later in digital download format.
It has beautiful packaging and a manual that spews "old school" from every pore.
Another great title for this 2023.

by **Carlo Nithaiah Del Mar Pirazzini**

OUR FINAL SCORE

» Gameplay 98%
Beautiful and dynamic gameplay. Well-structured levels and many in-game goodies. The two-button pad adds better maneuverability than the joystick on C64.

» Longevity 95%
There are 30 levels and they must be tackled with courage and accuracy. Not impossible, but definitely well paced.

NEW GAME

ELECTRONIC DREAMS THE SPACE ODISSEY

Year: 2023
Editor/Developer: Mikgames
Genre: Action game
Platform: Sega Master System
Website: https://www.smspower.org/forums/19827-ElectronicDreams

It is 2123 and you live on planet Earth. A group of terrorists attack the heart of Capital City, killing your father in the massacre.
You are a warrior and you are ready to avenge him.

Your journey will be throughout the solar system, from your home to the moon Sirius. You will do everything to complete your revenge.

A fresh new title for Sega Master System that pays homage to so many other games.

You'll find a bit of Contra and Metroid with lots of internal minigames and even a really well-done multidirectional shoot em up level.
The main character is a girl, initially armed with her repeating gun.

He will have to make his way through the levels among enemies, traps and extreme platform jumps and can also collect additional weapons and magazines to progress more easily.
Each level has its own well-crafted boss that must be taken down with careful observation of combat patterns.

One pad key is for firing and one key is for jumping. The PAUSE key functions as a menu for selecting weapons collected during the adventure.
This is a title that can be tackled in normal mode or in "easy" mode. The latter does not have the character's energy level and ammunition.

The game is quite challenging and traps the inattentive player.
I liked the graphics with the classic

Sega Master System palette. There is a lot of variety in the levels, several interesting animations, and it all moves without slowing down.

A beautiful soundtrack (which reminded me of Mega Man) accompanies us along carefully structured levels.

The choice to include multiple game modes (motorcycle race, metroid

OUR FINAL SCORE

» Gameplay 90%
Several game modes make it interesting.

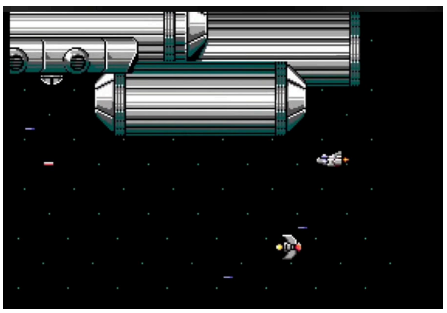
» Longevity 80%
Difficulty is perhaps too high. Controls not responsive.

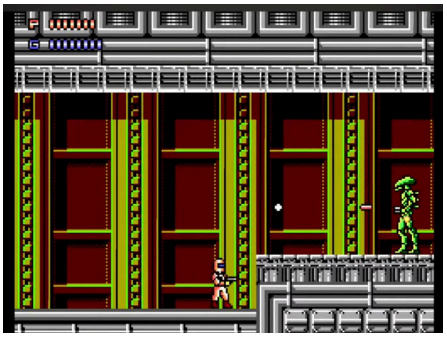
level, shoot'em-up level and the mini-games) make the title interesting.
Sore points: the controls are not as responsive as they should be, and perhaps one life is too little.

However, the developer is fixing it and promises that there will be future corrected versions.

Summing up, it is a good title for our Sega-branded 8bit.

by **Giampaolo Moraschi**





NEW GAME

RESHOOT PROXIMA 3

Year: 2023
Editor/Developer: Spieleschreiber
Genre: Shoot em up
Platform: Amiga AGA
Website: https://spieleschreiber.itch.io/rp3

After years of announcements and a long gestation (from 2019) here comes Reshoot Proxima 3.

Released after Amiga Germany 38 by developers Spieleschreiber, a collective already known for its strong presence in the demo scene.

A bombastic vertical scrolling shoot em up with impressive looks and ... a few problems I didn't expect.

The game appears in the manner of most classic titles in the genre: aliens to blast along 5 rather articulate and detailed levels (the second level in the sky is notable) with boss fights at the end of each level.

Our ship can be upgraded with two types of fire, a dual support pod and speed up (for a total of three effective upgrades) that also act as an "energy bar." Each time we are hit, we lose an upgrade until we lose life.

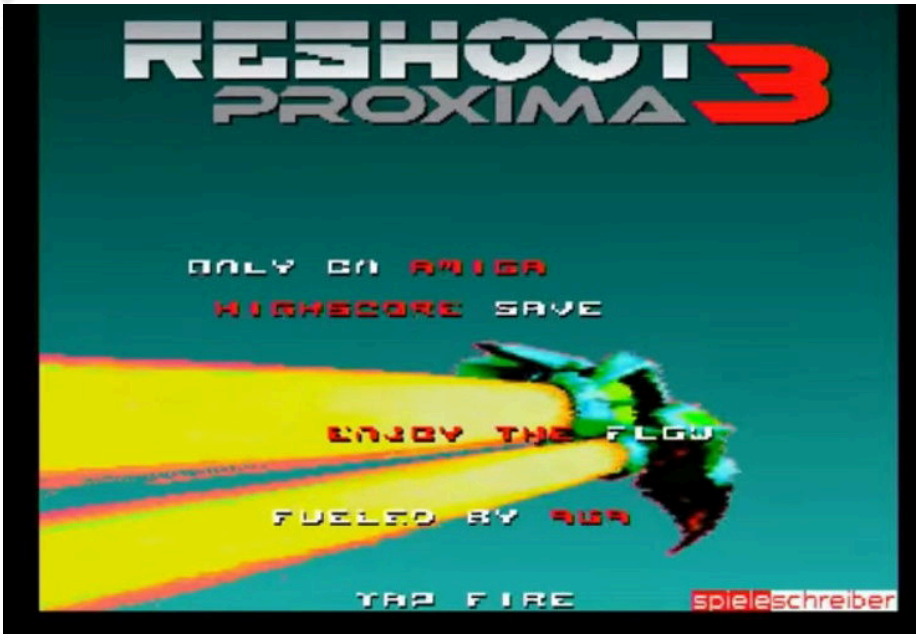
Hitting later leads to compos that accumulate points for the final score.

After a meager menu with options, the game confronts us with 5 different levels of varying length and good design.

It is definitely a title that winks at the

Each level is well characterized and features not only the classic waves of falling enemies, but also tests players' reflexes in untangling numerous traps.

Most important (and somewhat frustrating) to remember the number and exact location of opposing waves.



OUR FINAL SCORE

» Gameplay 80%

Well-structured levels but less-than-modern play style.
The second button as "Pause" is not a good choice.

» Longevity 80%

It is definitely a good game, although you have to remember every single wave in order to continue.

arcade shooters of yesteryear. It shares with many of them the difficulty and toughness.

Technically it is a gem. Colorful 24bit pixel art graphics that are super fast and super animated. Some levels are a feast for the eyes (the second level and the underwater one are incredibly beautiful).

Everything runs smoothly even in the simple basic A1200 where we tested the game without slowdowns.

Sound pumped up and well done. You can see the setting from Demoscene.

We come to the sore points.
It could have been the Amiga masterpiece of the new generation, but it is just a good game.

The structure of enemy assaults as we said is a bit archaic for today's times as is the impossibility of having only one type of fire and not being able to have the smart bomb (which is essential in some cases!).

While supported, the second button is limited to a simple button to pause the action. Too little and I personally find the choice irritating.

I hope for a future patch that can expand the game and improve the issues I have listed for you.

Reshoot Proxima 3 still remains a well-made and playable title that lacks that extra "bit" to be perfect.

by **Roberto Pirazzini**

NEW GAME

ROCKY & CO

Year: 2023
Editor/Developer: Icon 64 – Psytronik Software
Genre: Puzzle/Platform
Platform: Commodore 64
Website: https://psytronik.itch.io/rocky-and-co

The great explorer Rocky Memphis has discovered the entrance to the lost temple that he says contains the legendary treasure known as the GOLDEN CONDOR.

Also on the trail of this ancient artifact is his arch-enemy Russo Baluch, intent on getting his hands on it.

So, accompanied by two fellow adventurers, he sets off on an incredible hunt for hidden treasure. Rocky & Co is a puzzle platformer that reminded me of The Lost Viking and the Datasoft title The Goonies.

Each of the three team members has a different set of skills from which you can take advantage. The goal of each level is to use a combination of skills to get all three heroes to safety at the exit routes while avoiding any dangers on the way (and I assure you there are plenty of dangers!).

The player controls one character at a time. To change characters, simply click the Space Bar or alternatively click the Fire key for a while and press the lever at the bottom.
In solving all the various levels, there will be switches to stand on, handles to lower, and sometimes you will have to combine the skills of two characters to make something work.

The game is structured on 12 standard levels, but if you can light up the letters of the ICON inscription in the lower game panel you will unlock a bonus stage and a "special" ending. This game is a return of the protagonist on Commodore 64.

Borrowing the design of Datasoft's

OUR FINAL SCORE

» Gameplay 80%

Nice structured puzzles and game mode. I would have included a version for using the two keys in game. Also nice is the bonus levels option and the alternate ending.

» Longevity 80%

Twelve levels may seem few but I assure you that it will take you just the right amount of time to complete. The Practice option for new players is also interesting.
Mechanics a bit old school.

The Goonies, it is a perfect example of the Icon 64 team's ability to produce a modern product of high quality and surpassing the inspirational title. I find interesting the use of in-game characters and the design of some really well thought out and generally accessible puzzles.

Detailed and rich graphics. It really seems to jump off the screen. The soundtrack is by Saul Cross, also author of the previous Rocky Memphis title, perfect for the game and does a great deal to enhance the overall experience.

Rocky & Co., contains only 12 standard levels, but this does not make it a quick and easy game as the puzzles are structured at an increasing difficulty that will require a lot of skill and ingenuity. They have that Rick Dungerous element of trial and error that can turn the game into a real challenge (no continues and only 5 lives, mind you!).

Icon 64 to cater to casual gamers has included a practice mode that allows you to "work" on the style of play to begin the adventure. A wise choice that increases the longevity of the product by a few points.

It has some old-school mechanics and is definitely not super simple, but it is a good title.

If you like these kinds of games and love the platformer/puzzle combination then Rocky & Co is the one for you.

by **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

SNK VS CAPCOM

Year: 2023
Sviluppatore: Gianluca Alberico
Genre: Beat em Up
Platform: Commodore 64
Website: https://www.youtube.com/watch?v=fP0wBpNTXv8&ab_channel=RetroGL

It has been so many years that the Commodore 64 has kept us company. Years of unforgettable titles and endless afternoons spent in its presence. As endless as the games that continue to come out on the legendary "cookie."

This 2023, the C64's 41st year, has seen the release of such gems as L3STER, A PIG QUEST! and this SNK vs. CAPCOM.

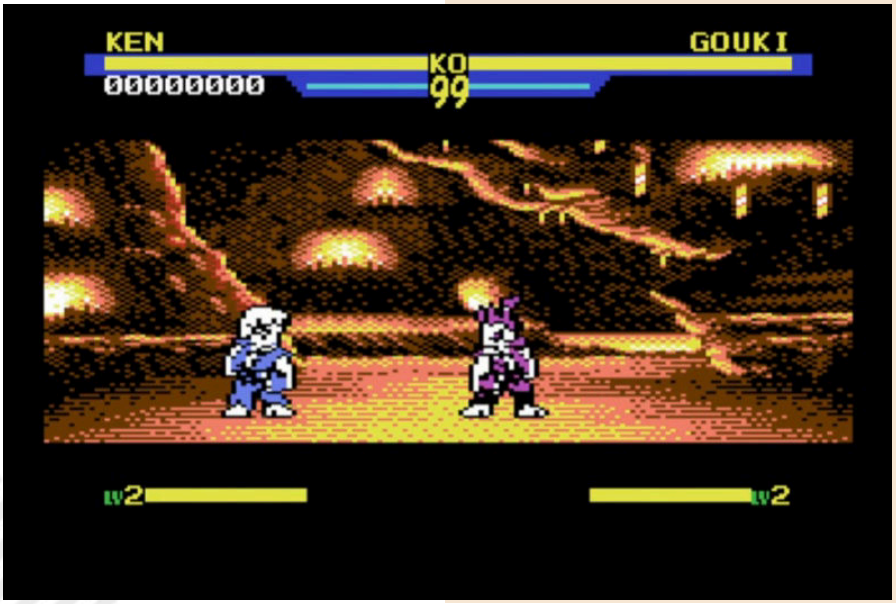
Author of the title dear Gianluca Alberico (accompanied by Jon Egglton). Originally started as a development engine for piacchiaduro on C64 and later evolved into a full game.

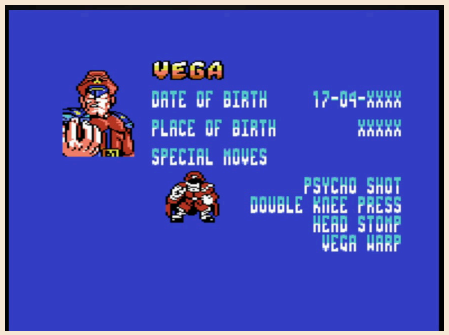
The title is inspired by the SNK vs. CAPCOM saga and in particular the title "MATCH OF THE MILLENNIUM" already admired on the unsuccessful Neo Geo Pocket Color. A monstrous programming, graphics and sound effort packed into a 1MB cartridge (CRT file). It will be difficult to see a version in other formats. It is a "fat" game and requires a lot of resources.

What do we see at the start of this title? Well, a nice array of options are present. There is the canonical Story Mode that will have you busy ridding the world of the villainous VEGA's Shadaloo (Bison for those who know him from the U.S. version) and his new ally G. Howard. There is the VERSUS version suitable for those who want to "take a beating" in the company and finally there is the interesting TOURNAMENT version with single head-to-head and elimination matches, quarterfinals, semifinals and tournament finals. This mode is also for two players.

We also have other options that allow us to select the difficulty (easy, normal and hard) and the possibility of using the joypad with second button support. Very welcome option the latter because it allows us to have separate kicks and punches.

In the game there are all the possible strikes: combo moves, parries, reverses, super moves and special





moves, and they are explained nicely in a nice little PDF manual of no less than 30 pages. Everything is there and it says how to do it. Super!

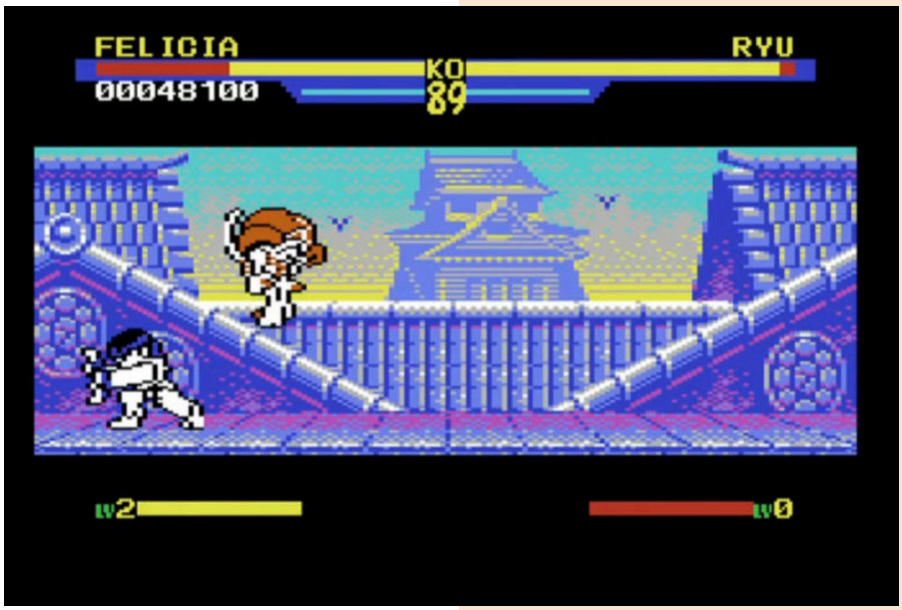
3 new fighters have been added since the previous Alpha release, 2 new bonus rounds have been added, and the artificial intelligence has been improved to increase the challenge.

We are looking at the best Commodore 64 fighting game ever! No ifs or buts about it. It has it all! Top-notch graphics with lots of character and stage animations and characterizations, a beautiful sound compartment, and perfect gameplay that leads to almost endless longevity.

Nothing is missing...and if that were not enough, it is free and open to future modification and improvement.

A masterpiece!

by **Carlo Nithaiah Del Mar Pirazzini**



OUR FINAL SCORE

» Gameplay 99%

Everything well balanced.
Perfect controls and the second-button option.
The three difficulty levels are perfect and the option of playing the game in three different ways make it appealing.

» Longevity 99%

Eternal like all those good fighting games of yesteryear.

LEGEND OF WUKONG

Year: 1996/2023
Editor/Developer: Gamtec, Super Fighter Team
Genre: RPG
Platform: Sega Megadrive
Website: https://www.legendofwukong.com/

A fine title released in China in 1996, translated into English in 2008 and finally reintroduced to the market in cartridge format as well, this Legend of Wukong (Wukong Waizhan in Chinese) is a role-playing game developed by Taiwanese Gamtec for Sega Megadrive (but also released on macOS and PC).

Its story is vaguely reminiscent of that of the novel Journey to the West, a Chinese tale published in the 16th century attributed to Wu Cheng'en and considered one of the greatest Chinese classics and the most popular literary work in the East. So popular that it spawned numerous cartoons, for example inspiring Akira Toriyama for the Dragon Ball saga.
The English translation was made by the Super Fighter Team, making the title more playable and usable outside the Far East.

The game is similar to many turn-based role-playing games of the 1990s. Battles are viewed from the side, and we simply plan our attacks, parries, use of cures and items for each character, and watch events unfold.
There is a lot of exploration and a lot to do in terms of in-game experience-everything you should expect from a classic role-playing game.

For me, what sets this title apart is the development of the story. Lots of interlude scenes explain the protagonist's journey and are combined with a series of dialogues full of witty and funny examples that will often make you laugh out loud. It's a different impact from the usual sad or melancholy plots of the standard role-playing games of the 16-bit era. I really enjoyed this freshness and lightheartedness of narrative during testing.
The game is well balanced. The battles, though quite frequent, are smooth and fair. There are no "U-turns" or paths where to get lost; if everything is done as the story commands, the game will flow smoothly and interestingly, engaging the player until the final confrontation.

The magic in the game is varied and very powerful, although it lacks "feature." Perhaps due to the simple animations the developers gave it. It does its job in the game, and that can and should be enough.

Remarkable work has been done on the dungeons. Although not huge in exploration they are well structured and challenge an inexperienced play

OUR FINAL SCORE

» Gameplay 90%
Good level design combined with a solid story and brilliant, entertaining dialogue.

» Longevity 90%
Well-structured dungeons, lots of exploration. A perfect mix for several hours of play.

party. I suggest entering the dungeons after leveling the play party a little bit.

The narrative, the gameplay, the design- all of this is done in a particularly fine way, with a subtle aesthetic appeal that grows more as you progress through the levels. The graphical look is classic but functional, and there is no need for a lot of parallax, scrolling, or wondrous lighting effects; what is there is perfect as it is. The cities and dungeons are colorful and detailed, and Gamtec has added little touches that are really well done.

A good soundtrack seasons the whole thing and fits the adventure. Each area has its own opening theme and changes on par with what appears to us as we progress through the game, always presenting itself differently.

The conclusion is that Legend of Wukong is a solid role-playing game with a refined style to add to the stable of this genre for our Megadrive.
The interaction between the heroes, the variety of enemies and environments, and the entertaining dialogues are a great way to spend time.
A title to rediscover that finally also finds a nice cartridge version complete with a well-done and colorful manual.

by **Carlo Nithaiah Del Mar Pirazzini**

NEW GAME

PAMPAS & SELENE

INTO THE MAZE OF DEMONS

Year: 2023

Editor/Developer: UnEPICFRAN

Genre: Platform

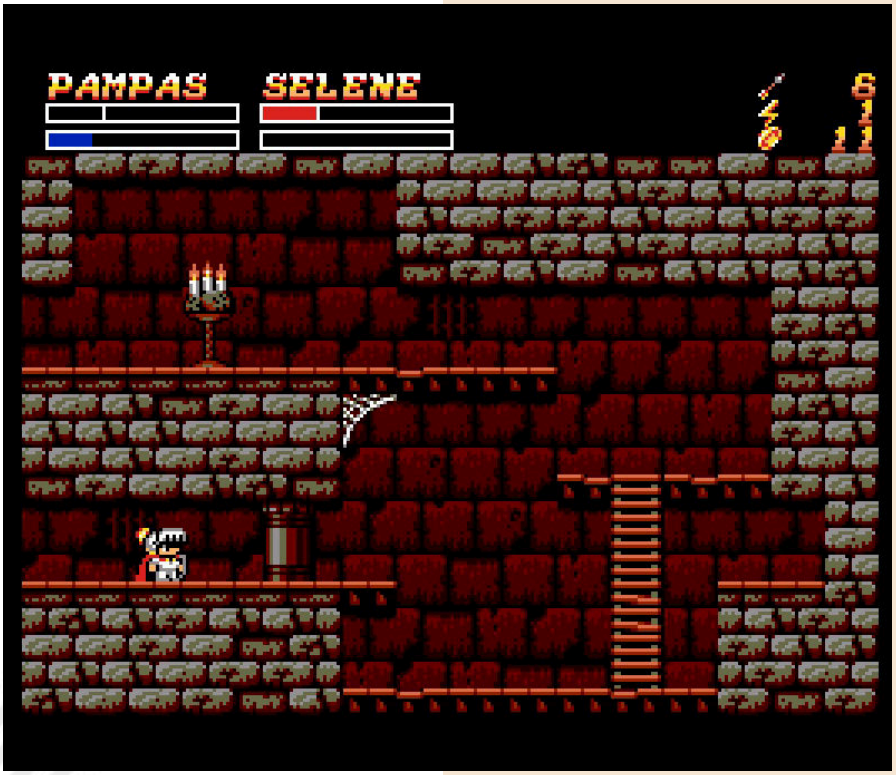
Platform: MSX 2

After defeating the terrible Galious, the heroic Popolon and Aphrodite returned to their daily lives. They lived a happy life as a couple in their cozy castle and raised their firstborn son Pampas, instructing them in the rules of knighthood.

Thirty years later a new enemy is on the horizon, and this time it will be up to young Pampas to rid the world of evil.

This title for MSX2 is the "spiritual" sequel to the famous Knightmare II-Maze of Galious that Konami made back in 1987 (itself a sequel to the beautiful Knightmare).

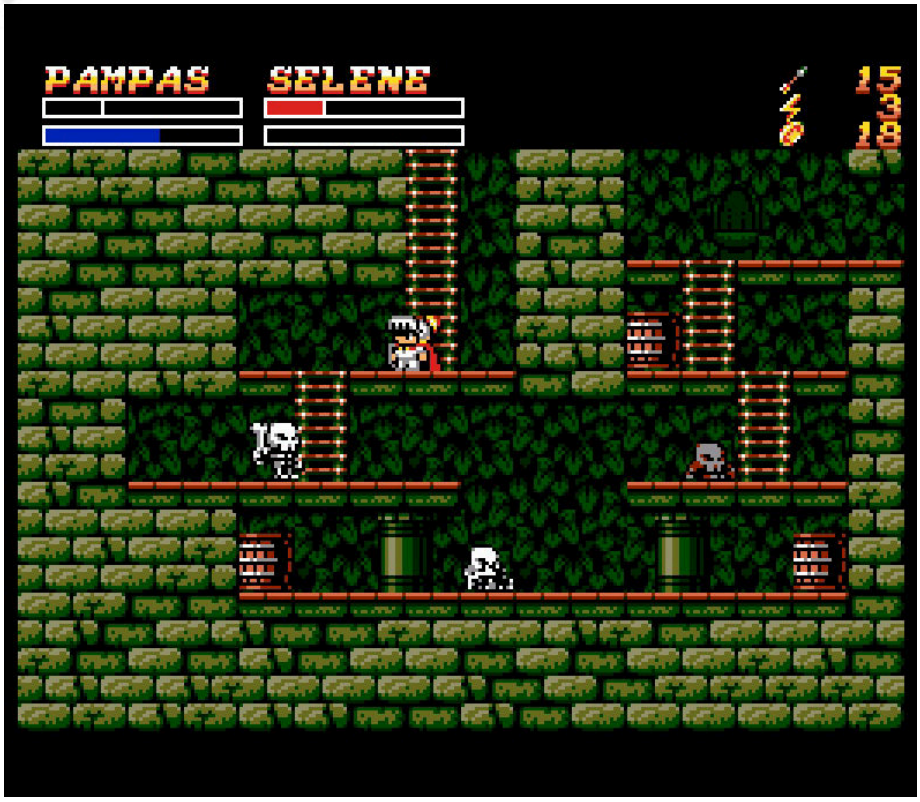
The gameplay is reminiscent of that of its famous predecessor. The player can choose whether to use the young hero or his little sister Selene, so he can use the special features of both to be able to progress through the more than 250 rooms chock-full of monsters, traps, and secret levels.



The game is very large and offers a considerable variety of exploration elements.

Remarkable and well-crafted title that uses screen 5 graphics mode for graphics.

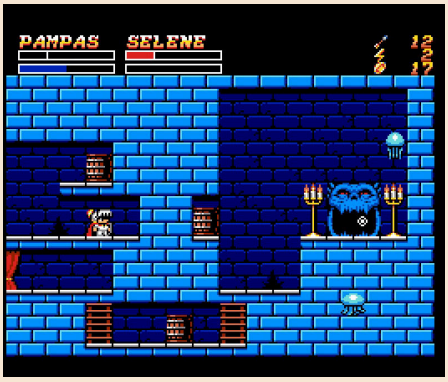
Technically it is undoubtedly a level product in every respect. The



OUR FINAL SCORE

» Gameplay 95%
Proven and entertaining gameplay.

» Longevity 98%
Challenging and VAST! It will take time to complete it all and that is good.

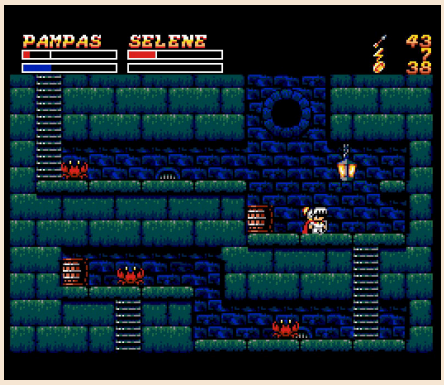


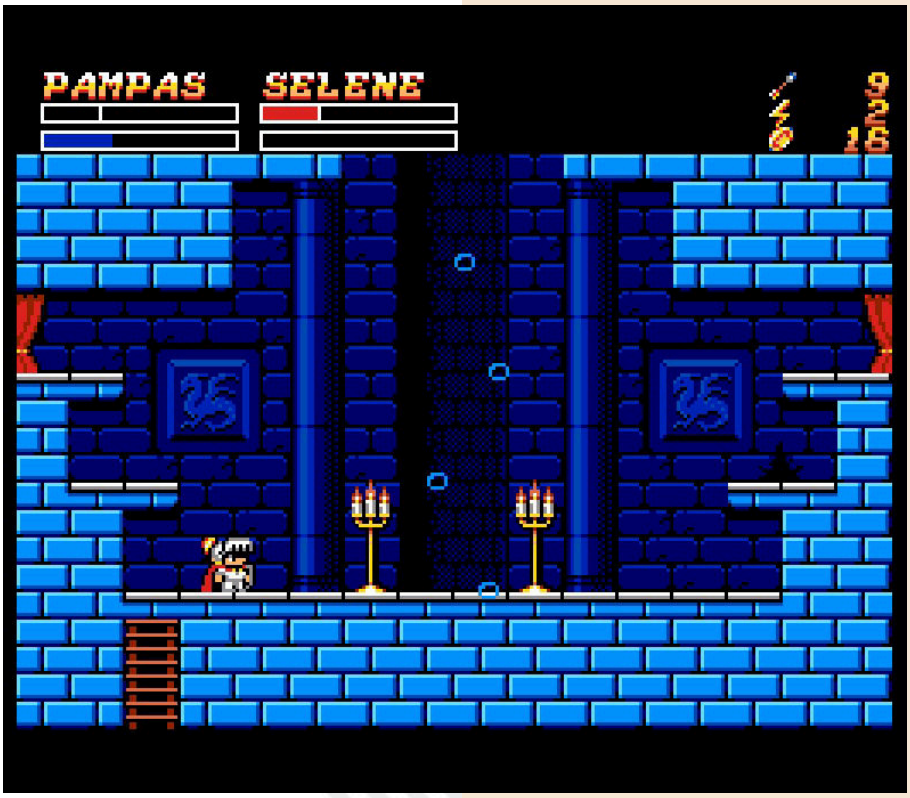
animations, the colors, the speed in the game--everything works perfectly.

It is a title designed as in the old days and requires manual dexterity in game mechanics.

Vast map and well-developed gameplay make this title one of the best MSX 2 titles ever.

by **Giampaolo Moraschi**





NEW GAME

STAR WARS: EMPIRE STRIKES BACK

Year: 2023
Editor/Developer: TCFS
Genre: Shoot em up
Platform: Plus/4
Website: https://plus4world.powweb.com/software/The_Empire_Strikes_Back

The force flows mightily on Commodore Plus/4! This is evidenced by the flood of titles recently released on this platform, all perfectly converted little masterpieces.

I like the Plus/4! It is one of those platforms made in a "weird" way but with great potential.

When it came out it was not even understood and was exploited badly, but thank goodness a handful of fans are astounding us insiders and all of you fans with wonderful titles.

In this case it is a conversion from a modern Commodore 64 title namely the Fan version of Star Wars Empire Strikes Back released in 2021.

If the version seen on the cookie made jaws drop this one is even more impressive when we think about some of the machine's hardware limitations. The little Plus/4 does very well with this title and does so by showing off

good game speed and great use of the TED sound chip.

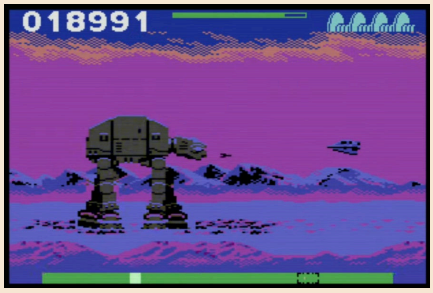
Technically as good as the C64 one but with the same problem.

Empire Strikes Back is basically a clone of Defender, a wonderful game but a child of its time.

In 2023, such a style of play could be boring and repetitive.

But who cares in the end! It is a good game and it is well made in this version of it, and it is undoubtedly worth trying.

by Giampaolo Moraschi



OUR FINAL SCORE

» Gameplay 95%
One of the most entertaining games of the golden age. Simple, straightforward, and fun.

» Longevity 90%
Increasing difficulty curve and a great desire to see how it ends.

NEW GAME

SUPER MONZA GP 2

Year: 2023
Editor/Developer: AJ Layden
Genre: driving game
Platform: VIC-20
Website: https://aj-layden.itch.io/super-monza-gp-2

An old-school subjective-view racing game made with passion by AJ Layden for all VIC 20s with 32K expansion.
The game expands the previous title by adding three new circuits (Monaco, SPA and Interlagos).
It moves fast, with beautiful graphics and priceless attention to detail.
Compared to the first chapter now cars receive damage from possible collisions and can be repaired by entering the pits during the race. More damage equals lower speed.
A championship mode has been added in which one must complete each race by getting on the podium in order to continue.
It runs on PAL and NTSC systems but is designed for the former. We have noticed that in NTSC mode it runs slightly faster.
Minimalist but effective sound.
The title lets you play the game well and entices you to go on to later circuits.
Highly recommended for all racing enthusiasts.
by **Giampaolo Moraschi**

OUR FINAL SCORE

» Gameplay 90%
Simple and well-structured.

» Longevity 90%
The Championship mode is an added point to the game's longevity.

NEW GAME

CORNELIUS
IN THE FORBIDDEN ZONE

Year: 2023
Editor/Developer: Physical Dreams
Genre: Platform
Platform: MSX
Website: https://www.facebook.com/profile.php?id=100064593073130

In a world of games with 4K graphics, ultra-realism, CGI, and everything more, we retro computer lovers are probably most excited to see hand-drawn images on the screen. True pixel mastery by real people. It is an art, whether using a real brush, pencil, or keyboard/mouse setup. To see a title on the MSX with graphics as curated as this Cornelius is a joy to behold and makes the heart flutter.
Inspired by the "Planet of the Apes" film series, the game Cornelius in the

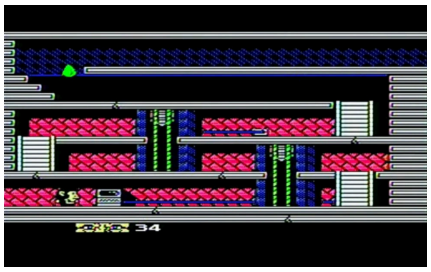
Forbidden Zone, is the latest production from Physical Dream, developer of new titles for MSX.

In the game we will guide the chimpanzee Cornelius through the forbidden territories where terrible--human beings live!
The game follows the classics of this genre. You jump over enemies, avoid traps, and try to arrive helpless (with all your energy bar full) at the end of the level.

The game action unfolds smoothly, and the levels grow in difficulty with a fair measure.

It may not be the most original of the titles on this game system, but it is definitely fun and from a technical standpoint well developed.

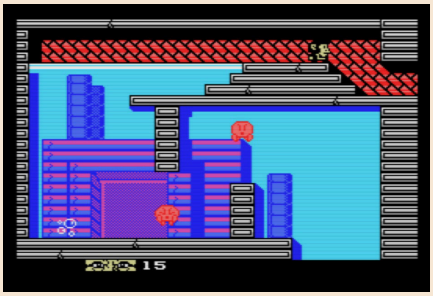
The simple graphics are as appreciable as the sound, punchy and well done.



The title is available in English, Spanish, and Italian and is also available in a physical version by contacting the folks at Physical Dream via email.

The cost of the title will be about 30 euros, a decidedly honest price for a fun title.

by Giampaolo Moraschi



OUR FINAL SCORE

» Gameplay 85%
Simple but well done, lets you play without too much to think about.

» Longevity 80%
It's not super long but it's fun.

NEW GAME

TIME WIZARD

Year: 2023
Editor/Developer: Amarok
Genre: Platform
Platform: Atari XL/XE
Download: https://forums.atariage.com/topic/354234-time-wizard-a-game-for-abbuc-software-contest-2023/page/3/

First review in RetroMagazine World and first title for a machine that has always fascinated me.

When I was growing up, my father had a computer store full of all these strange acronyms for me then: "Atari," "Commodore," "Spectrum." I played little video games, however: I was more interested in Barbies.

It wasn't until I got older that I became interested in retrogaming, gleaning through flea markets and the Internet the things that intrigued me the most. The 8-bit Ataris are incredible machines. They always showed colorful and beautiful games, and I recovered a computer a year or so ago from a friend. Too bad it is still in need of repair.

In the meantime, I'm going to try playing in emulation (I'm aware that it's not the same, but alas, I have to make do for now!).

Time Wizard is a platform game with a plot that I would call mundane.

Evil robots want to destroy the world by changing its history. We play the Time Wizard, the only bastion left to defend humanity.

We will be sent into the future to prevent disaster.

The task will be to collect the hourglasses of time so that the robots cannot turn back.

Time can be stopped even during the game if we make a mistake.

During the game, one can press the fire button on the pad to initiate time manipulation. Holding it down, moving to the left allows you to rewind, while moving to the right restores. A curious option that got me so excited.

On the introductory screen you can select one of 15 levels. At the beginning all but the first level are locked.

You must successfully pass the level to unlock the next one.

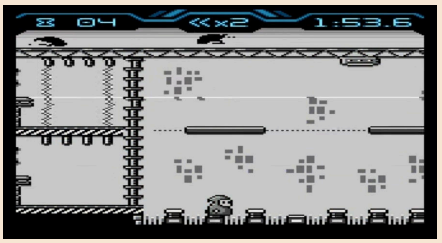
The game participated in the ABBUC Software Contest 2023, an event for Atari developers, placing first place. For me, a well-deserved position.

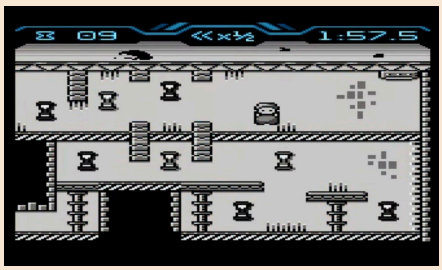
It is fun, has a well-structured and thought-out game concept, and the work the developer did using MADPascal, a 32-bit Turbo Pascal compiler for Atari XE/XL, is excellent. The graphics are colorful and fast-paced, and the soundtrack (which makes extensive use of classical music) is beautiful.

I am glad I started on RMW on the right foot because of this title.

The game runs smoothly in emulation with a configuration with at least 128KB of RAM.

by **Marta Rossmann**





OUR FINAL SCORE

» Gameplay 92%
Beautiful time manipulation mechanics and technical setting.

» Longevity 90%
The game mechanics and level setting will keep you engaged.

NEW GAME

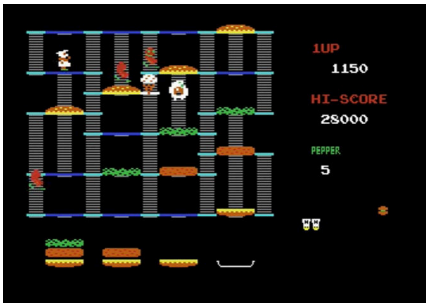
BURGERTIME

Year: 2023
Editor/Developer: Data East/ Arlasoft
Genre: Action/Arcade
Platform: Commodore 64
Website: https://arlagames.itch.io/burger-time

Peter Pepper on Commodore 64 has never been in better shape!

Of course, he still looks the same as he did in 1982, but in this new appearance on C64 he looks splendid and resplendent more than ever.

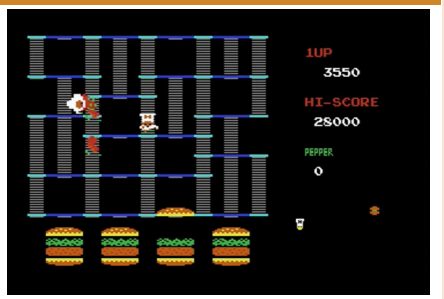
Data East's wonderful, multi-converted title for DECO Cassette System comes thanks to Arlasoft in a virtually perfect version.



All the animations are in the right place, and the spirit of the coin op is reproduced with great care.

The title, beloved by generations of gamers, needs little introduction.

Quick review: the aim of the game is always to complete levels consisting of ladders and shelves on which huge layers of hamburgers lie. The layers are arranged vertically, so that they fall directly on top of each other.

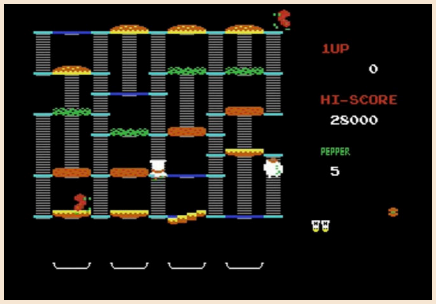


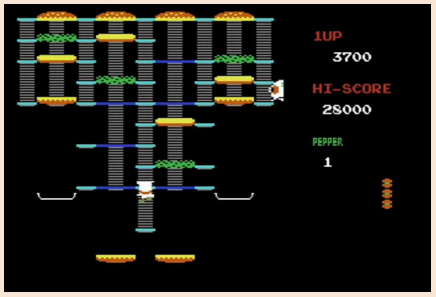
The player guides the chubby cook Pepper, who will walk over the layers and drop them down, thus composing the complete sandwich. All while evading the wacky enemies found in the levels-Mr. Hot Dog, Mr. Pickle and Mr. Egg.

The title can be downloaded from here: https://arlagames.itch.io/burger-time *and is worth collecting in our toy library.

Immortal!

by Carlo Nithaiah Del Mar Pirazzini

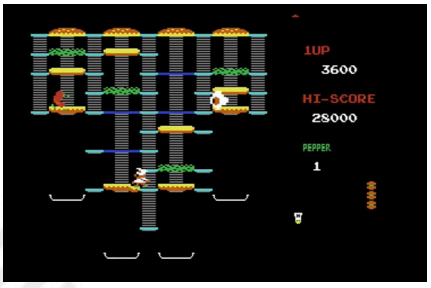




OUR FINAL SCORE

» Gameplay 90%
A classic of the genre perfect in this version.

» Longevity 90%
Again, if you get hooked you will hardly stop.



DRACULA'S CASTLE

Year: 2017
Editor: MP Software
Sviluppatore: Marco Pistorio
Genre: Platform
Platform: Commodore 64

This Halloween, too, ended under the banner of video games, soundtracks and images old and new thanks to the sixty-fourers who, despite everything, in 2023 are reliving those golden years and have never said no to their (and our) passions. I was convinced I would pick up a trite game this year, maybe even already reviewed in dozens of magazines, but no! A few days before Halloween fate would have it that I saw a gameplay of a game only six years old and unnoticed by many.

It's Dracula's Castle, a fixed-screen platform game in which the protagonist, the resident ghost hunter, finds himself in a house haunted by bats, vampires, and Count Dracula himself. Released in 2017 thanks to the diligent work of our retro enthusiast friends (in this case our fellow editor Marco Pistorio, ed.) who still program video games for 8-bit machines, in a slightly modern style but still worthy.

All we have to do, besides avoiding enemies thirsty for our blood, is to retrieve a key that allows us to open the door and reach the next room, and once we have passed three rooms we will finally reach the castle exit. The whole thing is not comparable to a school trip to a nineteenth-century castle; here the vampires are real and it will not be easy to get to the end by avoiding them as if they were people you do not like.

Fortunately, there are crosses that make us invincible for a short period of time. I said well, brief! Short could also be the game with only three rooms; many might observe that hardly a huge castle would contain only three rooms... The game's longevity is definitely centered on simple gameplay and suspense dictated by the soundtrack: toccata and fugue by Johann Sebastian Bach.

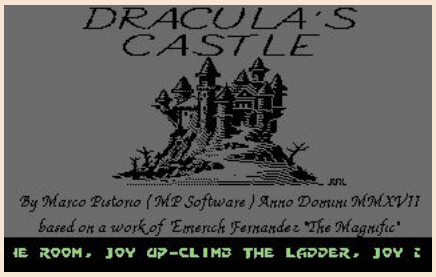
The graphical compartment is very reminiscent of that of the 1980s and features sprites a small, but still faithful to the characters. The strategic component of reaching the key, using invulnerability crosses, and outflanking enemies will be critical to getting through the game. There is only one flaw, which is the enemies that suddenly pop out from the left or right side of the walls; however, nothing major and you certainly won't need endless patience to finish it.

This beautiful title is not only to be played during the most monstrous night of the year, but whenever you want to have a quick game even in the middle of summer or during a night storm, in short, a title that will certainly never gather dust.

I don't think we need to forire advice to finish the game, the only thing I can tell you is enjoy it! We have just entered the cold season, the one with cover, hot tea, TV series and last but not least, retro video games!

Nothing like a cold, drizzly evening, touching Bach and Dracula's Castle!

by Daniele Brahimi



OUR FINAL SCORE

» Gameplay 80%
Simple, linear control.

» Longevity 50%
Only three levels, but suggestive and never repetitive!

New ideas for our future!

A lot of contents in this issue, the 23rd of RMW English version!
Hardware, software, programming, listings, interviews, game reviews, books, you name it. To date we can say, without fear of contradiction, that RetroMagazine World is one of the most comprehensive magazines in the retrocomputing scene.
When it all began six years ago, we would never have believed that we would get to publish 45 issues in Italian and as many as 23 in English.
And all this work, it is always worth remembering, always comes to you in a free form, because we at RMW are driven only by our passion and not by the need to make money; and this actually puts us in a privileged position!

We have no deadlines to meet, we are accountable to no one, and as a result we are under no pressure and no sword of Damocles hangs over our heads.
What does it all mean?
It means that we can document, experiment and develop the topics we ourselves choose, without any imposition or time constraints that might rush or degrade our work.
If an article is ready, we publish it. If it is not, we will publish it in the next issue or when the author feels the time is right to do so.

Why do I tell you all this? Because I want to make you understand the dynamics of our editorial staff and show you how editors are masters of their time and manage articles with total autonomy.
This kind of 'anarchy' has been working since the beginning of our adventure, and I think it's kind of our trademark.

Without wishing to appear presumptuous, somewhat like in academic circles, where young researchers are left free to experiment and follow their own intuitions, fueling the creativity of the whole group, everyone here is free to suggest and propose new ideas.

A couple of these new ideas are taking hold in the very pages of this issue. Let's see if you are able to identify them.

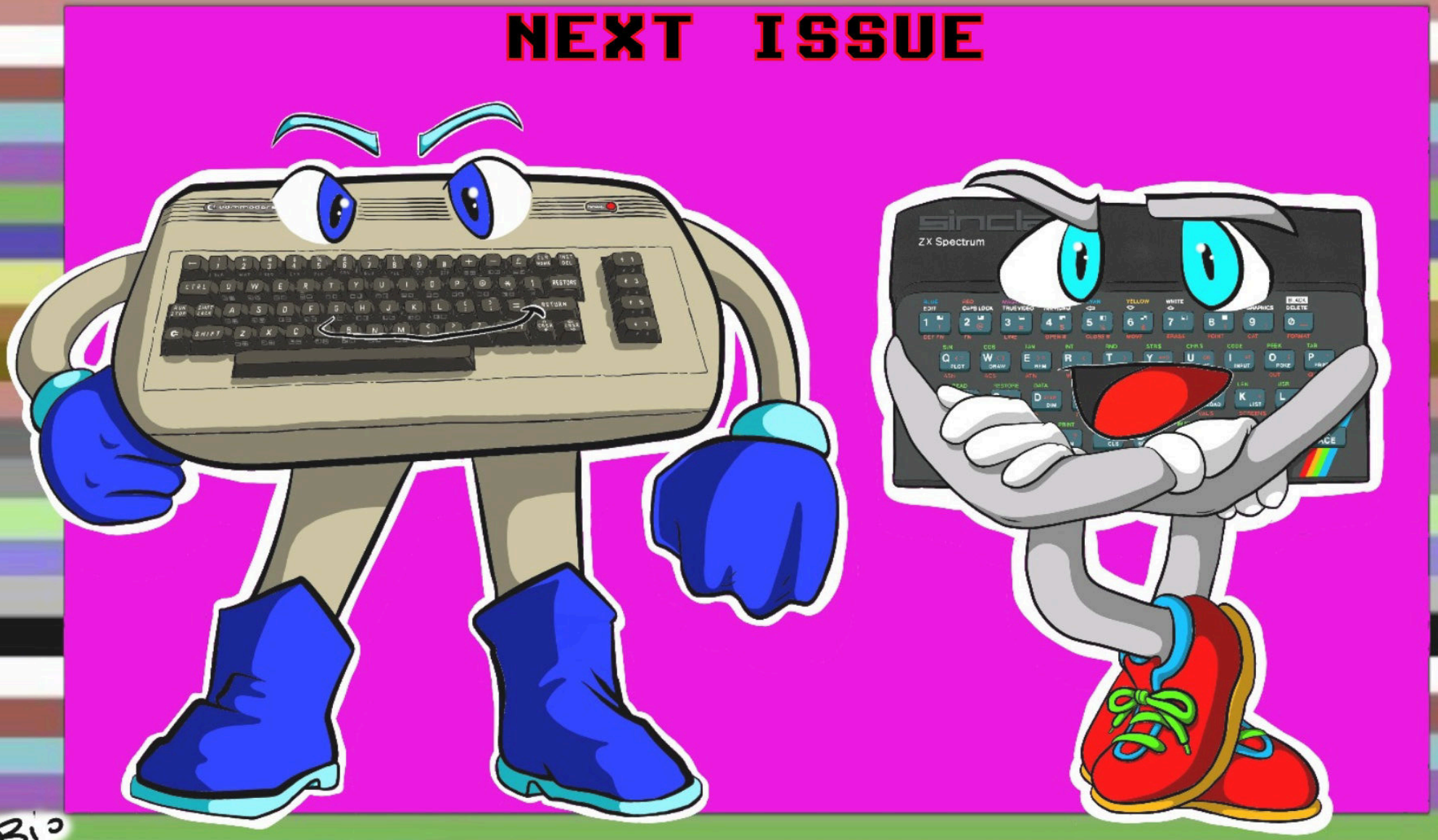
How about you? Do you have any interesting ideas or insights to suggest?
Come forward, we are always ready to listen to you and, if possible, put your suggestions into practice!

Francesco Fiorentini

RetroMagazine World
Year 4 - ISSUE 23 - OCTOBER 2024

Editor in Chief
Francesco Fiorentini
Editing/Deputy Manager
David La Monaca
Publishing Manager
Marco Pistorio
Website/Social Media Managers
Giorgio Balestrieri/Carlo Nithaiah Del Mar Pirazzini

SEE YOU IN NEXT ISSUE



THE DYNAMIC DUO!

FABIO DE RENEIS '23

RETROMAGAZINE WORLD