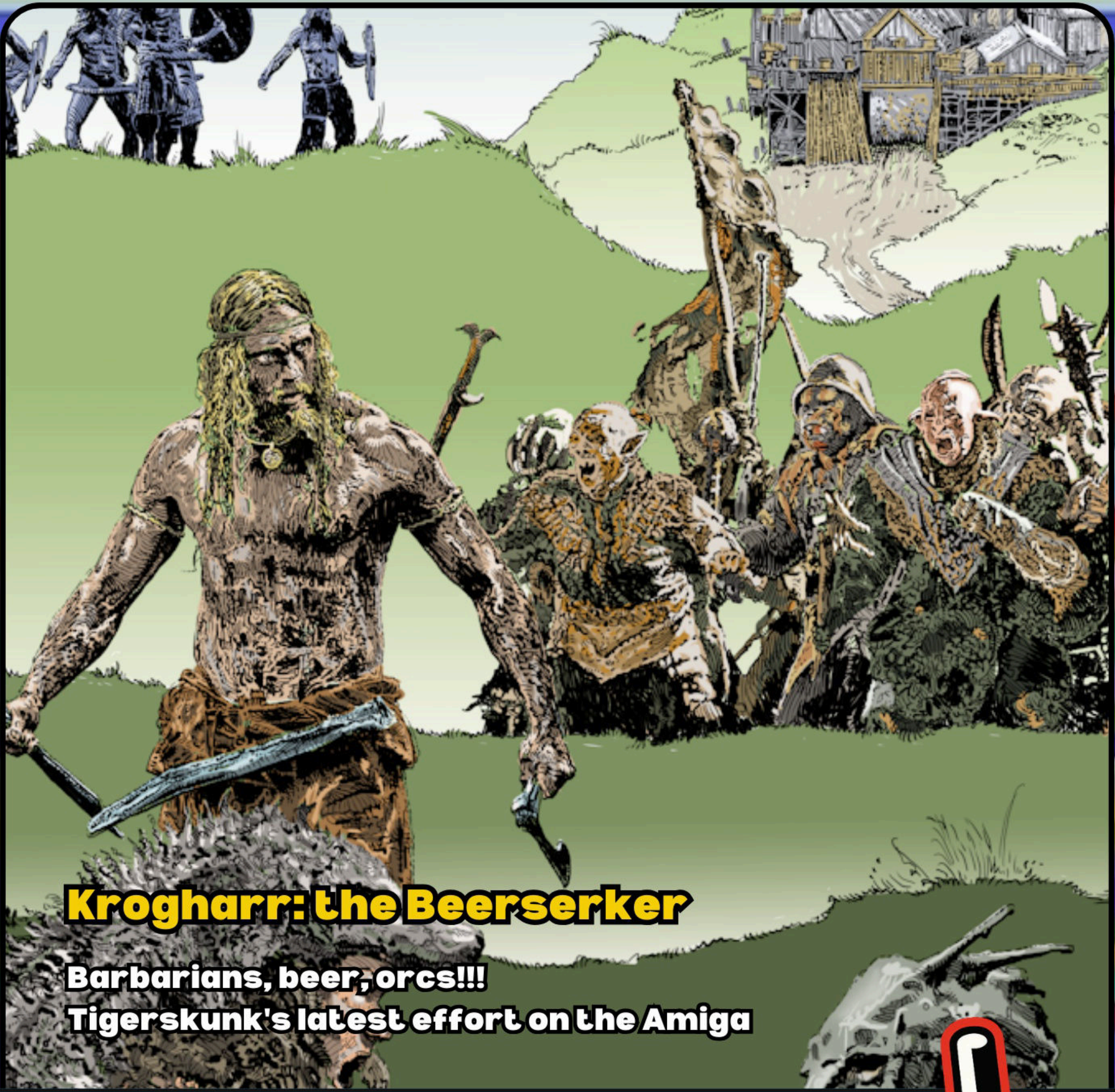
PROGRAMMING - HISTORY - INTERVIEWS - REVIEWS - INSIGHTS

WORLD RetroMagazine

FUTURE DAYS ARE BACK

ISSUE 29



Krogharr: the Beerserker

Barbarians, beer, orcs!!!
Tigerskunk's latest effort on the Amiga

All about the NEW COMMODORE

- Commodore 64 Ultimate
- interview with Peri Fractic

www.retromagazine.net

Why retrocomputing is more relevant than ever!

I'm inspired by a couple of discussions I've had recently. In an age where artificial intelligence reigns supreme and our data is stored "who knows where" in the cloud, there is a growing paradox: the fascination with computers of the past. We have talked about the nostalgia effect ad nauseam, but personally I don't think that's all there is to it. That would be reductive. So I asked myself a couple of questions and tried to answer them.

A return to tangibility

In a hyper-technological world where everything is dematerialized, we feel the need for tangibility. Computers of the past were tangible, accessible, even "humanly" imperfect. You could open them, understand them, modify them. But it's not just that. Today's games are almost exclusively in digital format. Once purchased, we can download them from somewhere (sometimes not even that), but we are left with nothing in our hands. No physical media, no manual to take to bed and fantasize about future games, no box to display on the shelves or in the bookcase. Personally, I miss that. In a few years, what will today's young people have left of the games that accompanied their childhood? Nothing...

Rediscovering limitations

More and more coders, who are certainly also experts in modern languages, are rediscovering the charm of creating software for machines with significant limitations (a few kilobytes of RAM, slow CPUs, very low video resolutions...). Probably the challenge of managing to create something relevant despite the objective limitations restores a sense of mastery and creativity that is often lost in modern development.

Sociological and cultural aspects

Recovering, restoring, and documenting systems, software, magazines, and books from that era means preserving the memory of a time when computer science was exploratory and visionary. It means treating computers not only as tools, but as cultural objects. Objects that, like it or not, have changed the social fabric and habits of millions of people over the last 40 years!

An active community

It is therefore clear that a global and, we hope, increasingly intergenerational community has formed around retrocomputing. Forums, social media, YouTube channels, open source projects, and dedicated fairs tell of a passion shared by more and more people. And while it is true that the majority are nostalgic family men, young people are also discovering the charm of machines and games that are light years away from modern glitz. My daughter (6 years old) regularly plays games on Amiga, C64, GameBoy, and Nintendo DS...

Retrocomputing is therefore not just nostalgia: it is also awareness! It is the desire to understand how and why we got to where we are now. And perhaps it is precisely this awareness that can generate curiosity even in the youngest, offering them not only a lesson in digital history, but a genuine spark of passion. I hope so. What about you?

Francesco Fiorentini

SOMMARIO	PAG.
◊ ...Commodore 64 Ultimate	Pag. 3
◊ Happy Birthday! 40 Years of Amiga	Pag. 6
◊ Happy Birthday Atari ST... The JackIntosh!	Pag. 8
◊ Commodore rises from the ashes (again)	Pag. 10
◊ ...The new version of the ZX Spectrum NEXT	Pag. 12
◊ The MSX that never was: a forgotten prototype	Pag. 13
◊ Visual Basic 6 on the browser?	Pag. 16
◊ Rotating octahedron on the Commodore 64	Pag. 18
◊ The Retromagazine World Library	Pag. 35
◊ The mysterious commands of MS-DOS: EDLIN	Pag. 36
◊ Random permutations for a question	Pag. 38
◊ Dario's books, AI, and the C64	Pag. 42
◊ Generation 80	Pag. 45
◊ The "new" Commodore	Pag. 46
◊ Interview with Luigi Bonifacino	Pag. 49
◊ ...Gachagachagachagacha...pon!	Pag. 54
◊ The unknown story of Takeru...	Pag. 58
◊ Kroghar (Amiga)	Pag. 60
◊ Outrun Amiga Edition (Amiga AGA)	Pag. 62
◊ Shantae Adavnce (GBA)	Pag. 64
◊ Galactic Panic (Atari ST)	Pag. 66
◊ The Cursed Legacy (MegaDrive)	Pag. 68
◊ Lunar Skirmish (SMS)	Pag. 70
◊ The Secret of the Four Winds (SMS)	Pag. 72
◊ Knight of the Moon (NES)	Pag. 74
◊ Tiny Magic (MSX 2)	Pag. 76
◊ Earrthion (Varie)	Pag. 78
◊ Cubix (Spectrum)	Pag. 82
◊ Daemonclaw: Origin of Nnar (MegaDrive)	Pag. 84
◊ Gladmort (Varie)	Pag. 86
◊ P47 II MD (Megadrive)	Pag. 88
◊ Rust 'n' Steel (C64)	Pag. 90
◊ Baten Kaitos (Gamecube/Switch)	Pag. 92
◊ Boni in Bat Hell! (Spectrum)	Pag. 95
◊ El Cartero (C64)	Pag. 96
◊ Track 'n' Field (Amiga AGA)	Pag. 98
◊ Outrun Game Boy (Gameboy)	Pag. 100
◊ Random Ancient Stones (C64)	Pag. 102
◊ ZPF (MegaDrive)	Pag. 104
◊ Tiger-Heli (Atari 7800)	Pag. 106
◊ Castle of Terror (GBC)	Pag. 109
◊ Jixa Ledy Tiger (Windows)	Pag. 110
◊ Dude the Deck Swabber (NES)	Pag. 112
◊ Defender (Atari ST)	Pag. 113
◊ Gyruss (Amiga)	Pag. 114
◊ Donkey Kong (Plus 4)	Pag. 115
◊ Phil's Adventure (Plus 4)	Pag. 116
◊ Willow (Arcade)	Pag. 117
◊ Conan (C64)	Pag. 118
◊ Hokuto No Ken (Playstation)	Pag. 119
◊ Speed Freaks (Playstation)	Pag. 120

- Cover image:
  Giuseppe Mangini
- Cover layout:
  Carlo N. Del Mar Pirazzini

Publication Note: all the contents of this issue were first released in Italian in September 2025

A look to the future
The Commodore 64 Ultimate

by Carlo Nithaiah Del Mar Pirazzini

The Commodore 64 Ultimate is the official reissue of the famous personal computer more than forty years after its release.

It is certainly an ambitious project that combines historical fidelity with a series of hardware enhancements designed for all retrocomputing enthusiasts and (hopefully) a new generation of users.

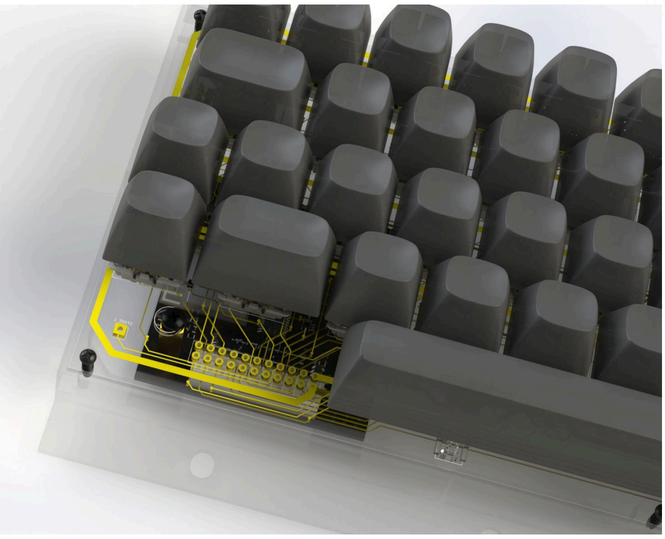
This is not a software emulation. The C64 Ultimate was conceived and built around an FPGA architecture (based on Gideon Zweijtzer's well-known 64Ultimate motherboard), a replica that faithfully reproduces the behavior of the original C64 motherboard, ensuring maximum compatibility and future expandability.

The heart of the system is an AMD Xilinx Artix-7 FPGA chip, supported by 128 MB of DDR2 RAM and 16 MB of NOR flash memory. The graphics compartment is capable of producing video signals in 1080p at 50 Hz (PAL) or 60 Hz (NTSC), with HDMI certification and DVI compatibility, as well as analog output via DIN-8, which allows CVBS, S-Video, or RGB connections. Audio outputs include a 3.5 mm headphone jack and an optical S/PDIF port. On the audio front, you can use two original SID chips (6581 or 8580) thanks to dedicated sockets with automatic voltage detection and filters, or take advantage of SID emulation via UltiSID technology integrated into the FPGA. There is also a SID-TAP header.

The C64 Ultimate is designed to be compatible with all hardware and software from the golden age of the C64. The claimed compatibility rate is 99% with games, cartridges, and peripherals from that era. Included are a cartridge port, Datassette, floppy drive, and joystick/paddle (2 x DB-9) ports, as well as a 26-pin user port. A 100 Mbps Ethernet port is available, and integrated Wi-Fi connectivity allows for wireless software transfer. Power

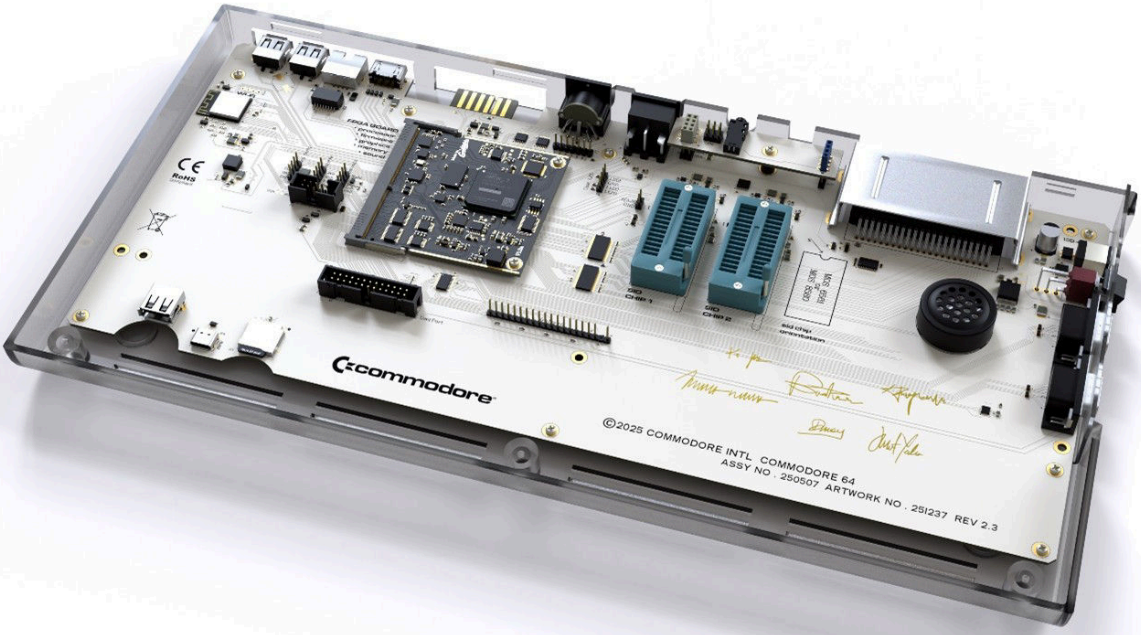
is supplied via a 12 V DC input, with a worldwide compatible power supply included in the box.

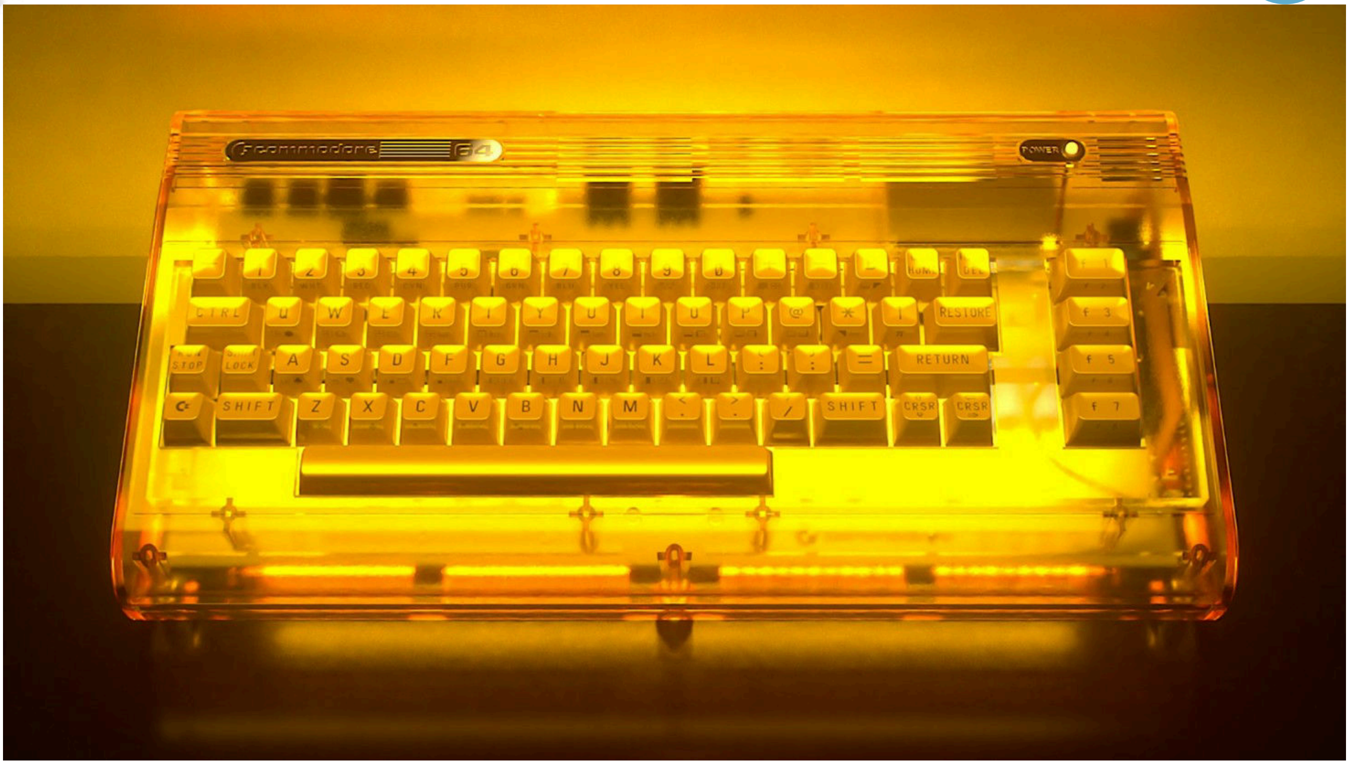
The system comes with a 66-key mechanical keyboard with pre-lubricated, PTFE-free 55 g Gateron Pro 3.0 switches, original C64 layout and shapes, full NKRO rollover for simultaneous keystrokes, support for programmable macros, and 70 RGB LEDs. The lighting, both for the keyboard and the case, is integrated into the motherboard and is fully customizable via the system menu, with control over patterns, speed, and brightness. The machine is compatible with FAT, FAT32, and exFAT formatted USB drives and supports a wide range of file formats including .D64, .D71, .D81, .G64, .T64, .TAP, .PRG, and .ROM. Included is a 64 GB cassette-shaped USB drive called "The Very Second," preloaded with over 50 full games, officially licensed classics, music demos, and the new exclusive sequel to Jupiter Lander, titled "Jupiter Lander: Ascension." The C64 ROMs are provided under license or can be configured via a wizard on first boot.



Distinctive features of the Commodore 64 Ultimate include copper-engraved signatures from some of the original and contemporary creators of the C64, including Albert Charpentier, known as the "father of the C64." There are aesthetic and functional elements designed to pay homage to the past, such as the ability to choose from different case variants, including the rare Founders Gold Label edition inspired by the millionth gold C64, featuring 24k gold-plated badges, gold-colored satin keys, translucent amber case, commemorative T-shirt, and Commodore-style "share" certificate.

The product package includes the computer, a spiral-bound user manual, quick start guide, HDMI cable, power supply with international adapters, sticker set, and comes

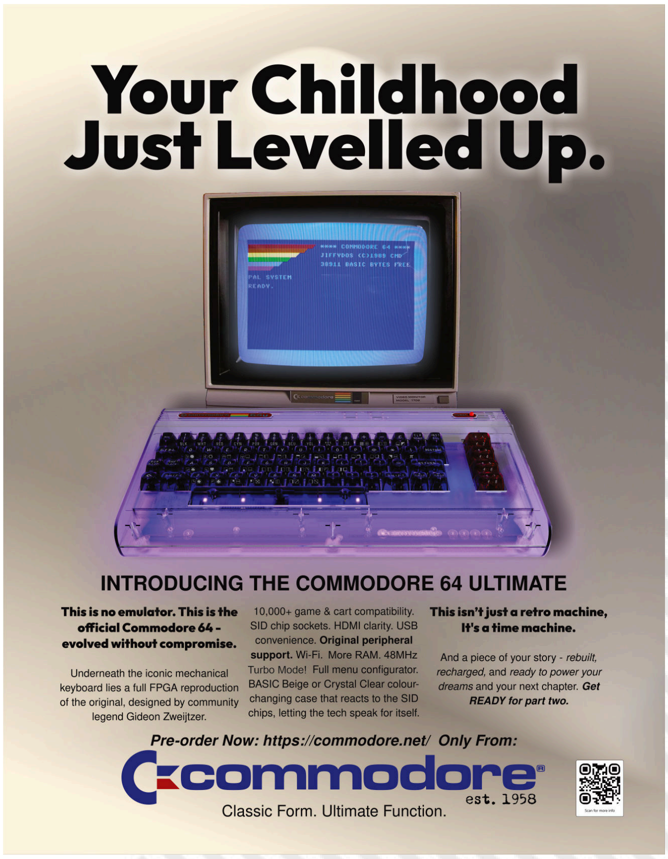

commodore
©2025 COMMODORE INTL COMMODORE 64
ASSY NO . 250507 ARTWORK NO . 251237 REV 2.3



in a glossy box in the original 1980s style. Shipments will depart from Commodore's logistics centers and its official partners in the United States, United Kingdom, and Europe.

The launch price is $349.99, currently on sale for $299.99 with a 10% discount if you buy two units. This is a direct pre-order and not a Kickstarter-style crowdfunding campaign, although the funds raised will go towards the production and relaunch of the Commodore brand. In this regard, the pre-order process does not involve any significant risks: most of the components are already in production, including cases and keycaps. Only minor elements remain to be completed, such as the keyboard base, which is being produced by experienced suppliers who have been active since the days of the Apple II. The design has been entirely conceived for integration between the various parts, including the manual, packaging, and power supply.

The pre-order amount is charged immediately in US dollars, while prices are displayed in local currency until checkout. A full refund policy is guaranteed until shipment: users can cancel at any time and obtain a full refund with a click from their account or through the support form. If customs duties or import fees are applicable, these are calculated and charged directly at the time of purchase under the heading "Tariff Tax," avoiding surprises upon delivery. If the fees decrease before shipment, the difference will be refunded; in the event of a significant increase, the amount may be updated, but only if strictly necessary to protect the sustainability of the Commodore project.



Your Childhood Just Levelled Up.

INTRODUCING THE COMMODORE 64 ULTIMATE

This is no emulator. This is the official Commodore 64 - evolved without compromise.

Underneath the iconic mechanical keyboard lies a full FPGA reproduction of the original, designed by community legend Gideon Zweijtzer.

10,000+ game & cart compatibility. SID chip sockets. HDMI clarity. USB convenience. **Original peripheral support.** Wi-Fi. More RAM. 48MHz Turbo Mode! Full menu configurator. BASIC Beige or Crystal Clear colour-changing case that reacts to the SID chips, letting the tech speak for itself.

This isn't just a retro machine, It's a time machine.

And a piece of your story - *rebuilt, recharged,* and *ready to power your dreams* and your next chapter. ***Get READY for part two.***

Pre-order Now: https://commodore.net/ Only From:

commodore®

est. 1958

Classic Form. Ultimate Function.

Scan for more info

Advertising flyer for the new Commodore

Happy Birthday! 40 years of Amiga

by Carlo Nithaiah Del Mar Pirazzini

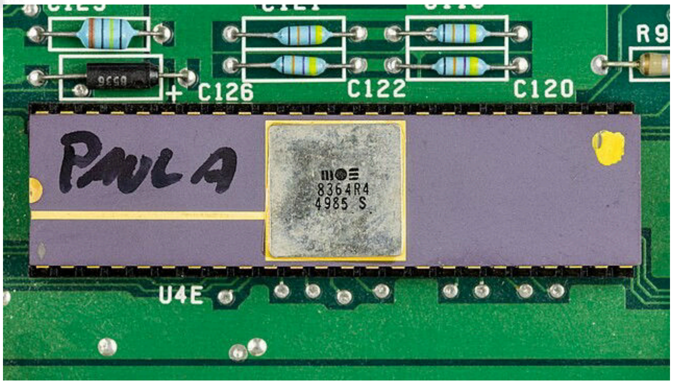
In July 1985, after three years of development, Amiga was presented to the public at a glamorous event complete with an orchestra, elaborate performances, and the participation of celebrities such as Andy Warhol and Dianne Harry.

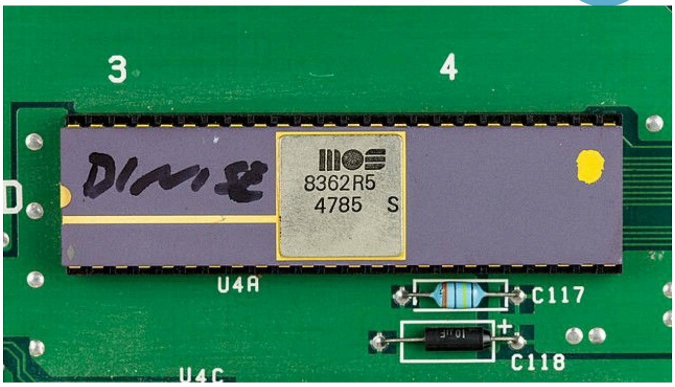
At a time when IBM PCs could only display a few colors and the Mac still had a black-and-white display, the Amiga (later renamed Amiga 1000 in its first incarnation) was revolutionary.
It could display up to 4096 colors simultaneously with HAM (Hold-And-Modify) mode, ideal for displaying photographs and 3D rendering. In standard mode, it supported up to 32 colors, far surpassing the 16 colors of the Atari 520ST, launched shortly before.

The Amiga wasn't just impressive for its graphics. It was capable of producing four-channel stereo audio, making it ideal for games and multimedia applications. In contrast, PCs of the time were still based on basic mono audio or even less, while the ST was equipped with the Yamaha YM2149F chip, which provided three audio channels plus a channel for "noise," which was good, but the overall audio experience was not as rich as what the Amiga offered. In addition, the operating system (developed by MetaComCo) supported full multitasking, allowing multiple programs to run simultaneously. This was a significant advantage over other systems of the time, which typically only supported single or cooperative multitasking.

The main reason Amiga could offer such advanced features was due to its use of the most economical consumer CPU, the Motorola 68000 (with a 16-bit external data bus but capable of 32-bit internal processing) combined with the Original Chip Set (OCS), which included three custom chips called Agnus, Paula, and Denise. These chips, designed by Jay Miner and his team at Commodore and manufactured by MOS Technology, handled most of the graphics and audio, allowing the CPU to focus on other tasks. This coprocessor architecture was a major step forward compared to other systems of the time, which relied entirely on the CPU to handle everything, including graphics, audio, and I/O operations.

Inspired by Apple, which signed the inside of the Macintosh 128K case, the Amiga development team decided to do



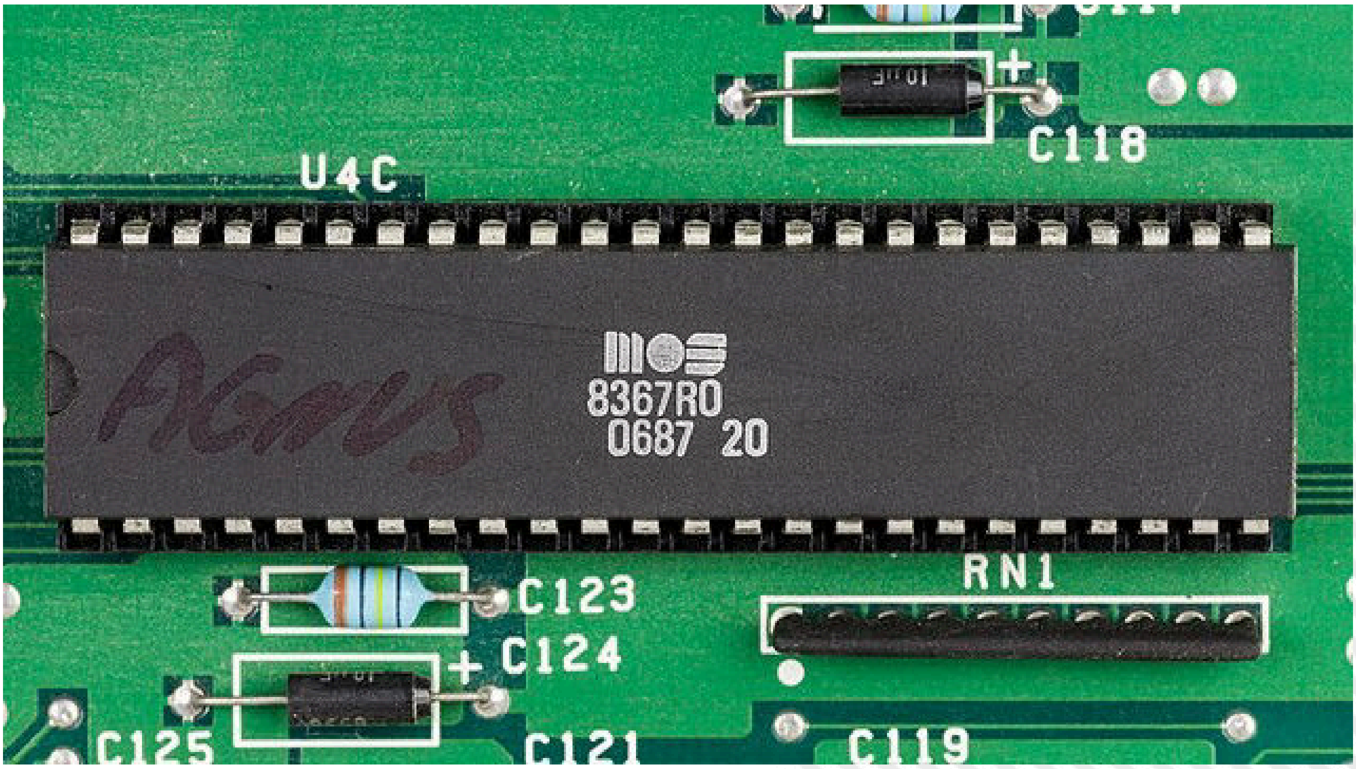


the same. The signatures of 53 members of the Amiga team were stamped inside the plastic shell of the Amiga 1000, but this detail is only found in Revision 6 (NTSC) and A (NTSC and PAL) models.

Two of these signatures stand out. The first is Mitchey, a paw print representing Jay Miner's beloved dog, often considered the team's "unofficial mascot." The second is Joe Pillow, a fictional character. When the Amiga team took the delicate Amiga 1000 prototype to the Consumer Electronics Show (CES) in January 1984, they had to secure a seat for it on the plane. The airline insisted on a name for the passenger, and the group jokingly chose "Joe Pillow."

Commodore launched an impressive promotional campaign before the Amiga's release. It became the most anticipated computer of 1985, but its commercial success was somewhat mixed. On the one hand, its coprocessor architecture, combined with its multitasking operating system, allowed it to offer cutting-edge performance for its time. On the other hand, developers had to learn new concepts, such as system resource management, in a different way than with other computers, making it risky to invest in this new platform, especially when established systems such as the Commodore 64 boasted a large user base and proven success. Therefore, most of the programs available for the Amiga were simply ports from other systems. The lack of exclusive software in the years to come, combined with strong competition (brands such as Apple and IBM dominated the market), hampered the commercial success of the Amiga despite its innovative design.

Unique, innovative, beautiful... Happy birthday, Amiga!



Happy Birthday Atari ST... The JackIntosh!

by Giampaolo Moraschi

1985 was not only the year of the Amiga, but also of the Atari ST. Introduced just a few months earlier (January 1985), the Atari ST was a highly regarded and commercially successful machine that is now celebrating its 40th anniversary.

Games such as Starglider, Dungeon Master, Populous, Defender of the Crown, Captain Blood... titles that I particularly loved on the ST.

You see, while the ST was something of a monster when it was launched in 1985 (there was a limited release between April and June of that year, with the wider launch coming in July), it was quickly overtaken by the Commodore Amiga, its direct rival, ironically produced by the company that Atari Corp. boss Jack Tramiel had just left in 1984.

Designed in just five months by a team led by former C64 developer Shiraz Shivji, the 520ST would later be joined by the STE, the STacy laptop, the TT030, the Mega STE, and the Falcon. The ST line was discontinued in 1993, and Atari would stake its future on the ill-fated 64-bit Jaguar console. Its name, ST, stood for "Sixteen/Thirty-two," referring to the 16-bit external bus and 32-bit internal components of the Motorola 68000 processor. It was sold with either the Atari color monitor or a less

expensive monochrome monitor. Color graphics modes were only available on the former, while high-resolution mode required the monochrome monitor. Most models could be connected to a TV in graphics mode. In Germany and other European countries, the ST established itself in the CAD and DTP sectors. Built-in MIDI ports made it extremely popular with amateur and professional musicians. The song "Your Woman" by White Town, which reached number one in the UK singles charts, was created using an Atari ST; in addition, artists such as Utah Saints, Atari Teenage Riot, Pet Shop Boys, Dario G, Fatboy Slim, Mike Oldfield, and Tangerine Dream all used the computer to create music at some point in their careers.

The operating system was GEM, previously developed for Intel architectures but later adapted for the 68000. Leonard Tramiel himself oversaw the project (renamed Project Jason) for the ST series. GEM was based on CP/M-68K, a direct port of CP/M to the 68000. By 1985, CP/M was becoming increasingly obsolete; for example, it did not support subdirectories. Digital Research was developing GEMDOS, a disk operating system for GEM, and discussed whether a port could be completed in time for the product's June delivery. In the end, it was decided to do the port, resulting in a GEMDOS file system that became part of Atari TOS (for "The Operating System," colloquially known as the "Tramiel Operating System"). This gave the ST a fast, hierarchical file system, essential for hard disks, and provided programmers with function calls similar to those in MS-DOS. The Atari ST character set was based on codepage 437. Forty years on, we remember another great machine from the past that is enjoying a second life, just like the Amiga. In fact, there are many retro game and software developers who are keeping Tramiel's dream alive. And Atari's. Happy birthday, Atari ST!

Commodore rises from the ashes (again)

by Carlo Nithaiah Del Mar Pirazzini

Christian Simpson, creator of the YouTube channel "Retro Recipes," has formalized the acquisition of Commodore Corporation, together with a group of former executives, to relaunch the legendary brand.

The Commodore brand is therefore enjoying a spectacular comeback under the leadership of Simpson, also known by his nickname "Peri Fractic."

In a recent series of videos, the YouTuber announced the acquisition of Commodore BV, the Dutch company that owned the remaining brands in the Commodore galaxy, including the famous original logo known as "chickenlips." After months of negotiations, this acquisition has been made official, marking a new milestone in the history of this legendary computer company.

An acquisition driven entirely by a passion for retrogaming. Christian Simpson announced that he had signed a share repurchase agreement with the previous Dutch owners of Commodore Corporation BV. The transaction was completed by a group of anonymous investors for a seven-figure sum, according to his own words. This is much more than just a commercial transaction, although the deal is not yet completely finalized: the company is still looking for investors to complete the transaction. The British YouTuber, who mortgaged his house to finance part of the operation, has surrounded himself with informal investors and an exceptional team of former company employees.

The group assembled by Simpson is striking for its historical legitimacy. Bill Herd, designer of the Commodore 128 and Plus/4, joins the project as a technical advisor and shareholder. Michael Tomczyk, former assistant to President Jack Tramiel and head of development for the VIC-20, brings his experience as a senior advisor. David Pleasance, former deputy general manager of Commodore UK in the 1980s and 1990s, takes on the role of asset consultant. The creative side of the project benefits from the involvement of Thomas Middleditch, a Canadian actor known for his role in the HBO series Silicon Valley, who now serves as creative director. A former Amiga user in his youth, Middleditch brings his artistic vision and humor to the brand's relaunch.

Ambitious plans for the future

Peri Fractic makes no secret of his ambitions for Commodore (you can read the interview in this issue). He outlines his goal of reviving Commodore with "retro-futuristic" products inspired by the 1990s and early 2000s, a time when computing still represented an optimistic ideal, in contrast to today's trends dominated by social media and artificial intelligence. This approach is already taking shape with the Commodore 64X, which runs on Vision OS 3, a proprietary operating system with a delightfully retro design. Simpson also reiterated his interest in Commodore OS Vision, a Linux distribution developed during the Commodore USA era by Leo Nigro, future technical director of the new organization.

The first commercial product of this new era, the Commodore 64 Ultimate, is now available for pre-order on the official website www.commodore.net. Based on the PCB of the Ultimate 64 Elite II (designed by Gideon Zweijtzer), this computer promises near-perfect compatibility with the original computer, while integrating modern features: modern video output, Wi-Fi connectivity, a real mechanical keyboard, and USB ports. Three versions are available, from the classic "Basic Beige" at â,¬257 to the limited edition "Founders Edition" at â,¬430.

A heritage to preserve and develop

The acquisition includes 46 trademarks dating back to 1982, providing a solid foundation for new product development. It should be noted, however, that this acquisition only concerns the Commodore trademarks. The rights to Amiga, as well as to 8-bit software and AmigaOS, are partly held by Michael Battilana (Amiga Corporation) following the bankruptcy of ESCOM. Simpson stresses the importance of creating an official licensing system that allows the community to showcase its creations, thereby democratizing the use of the Commodore brand. Public financial participation is not yet possible, as international legal constraints are currently too complex. In the meantime, Commodore is hiring a community manager and designer for its merchandising. The first product appears to have been finalized, but its announcement is reserved for a future video.

The project is not limited to commercial aspects. A Commodore Care fund will be created to ensure access to technology for disadvantaged children by installing Commodore computers in hospitals, schools, and homes. This philanthropic initiative is in line with Simpson's vision: technology at the service of humanity, rather than creating dependency.

Despite the enthusiasm generated by the announcement, many challenges remain. Full financing for the operation has not yet been secured, and Simpson continues to seek investors to finalize the acquisition. The complexity of the Commodore ecosystem, fragmented since its bankruptcy in 1994, also raises questions about the actual scope of the rights acquired.

This rebirth of Commodore, led by a retrogaming enthusiast, could nevertheless breathe new life into a brand that has marked the history of personal computing. With the Commodore 64 remaining the best-selling personal computer of all time, with 17 million units sold, there is still a considerable legacy to preserve.

Not only Commodore! The new version of the ZX Spectrum NEXT is coming soon

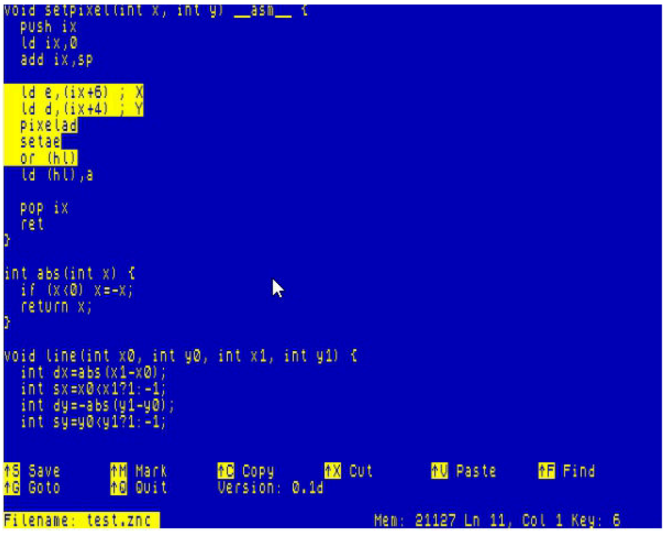
di Carlo Nithaiah Del Mar Pirazzini

The ZX Spectrum continues to evolve, not with a replica, but with an updated version called ZX Spectrum Next Issue.

This new version follows the success of its predecessors and introduces a larger and more powerful FPGA: the Artix A7 XC7A35T-2CSG324C, support for additional platforms such as the Sinclair QL and Commodore 64, and a host of new hardware and software improvements.

The NEXT is what the ZX could have been if it had continued to evolve: a machine capable of rivaling the Amiga or Atari ST, with hundreds of new games and apps, 256-color graphics in different modes, 9-channel stereo sound, fast storage, and much more.

Specifications:
- Processor: Z80N at 3.5MHz with turbo modes at 7MHz, 14MHz, and 28MHz
- Compatibility: Full compatibility with the ZX Spectrum family
- Memory: 2MB RAM
- Video: 2, 8, 16, and 256-color modes (high-resolution modes 128 x 96, 256x192, 320x256, 512x192, and 640x256)
- Video output: RGB, VGA, HDMI, 50Hz and 60Hz modes
- Extra hardware: Sprite hardware, zxnDMA, CTC, Copper, enhanced ULA, Level2 and Tilemap (Level3)
- Storage: 1 x SD Card slot, with DivMMC compatible protocol, 1 x internal microSD card slot with pre-installed card (fully supported by NextZXOS)
- Audio: 9-channel sound via three stereo AY-3-8912, plus two x 4-bit DACs
- Joystick: Two ports compatible with Cursor, Kempston, and Interface 2 protocols â€” extended to also support Megadrive joypads or configurable as serial ports
- PS/2 port: Mouse with Kempston mode emulation and/or external keyboard
- Special: Multiface functionality for memory access, saving games, cheats, and more
- Tape support: Combined MIC and EAR port for loading and saving
- Expansion: Original external bus expansion port (now with +3 disk interface support)
- Accelerator card (optional): GPU/CPU 1GHz/RAM 512Mb
- Network: Integrated Wi-Fi module
- Real-time clock: Integrated RTC with battery backup
- OS: NextZXOS and NextBasic, equipped with an expanded command set
- Alternative OS: CP/M (included), SymbOS, and esxDOS
- Additional cores and official ROMs: Sinclair QL/Next and C64 with cores that can also be launched via NextZXOS. Community cores available



```
void setpixel(int x, int y) __asm__ {
  push ix
  ld ix,0
  add ix,sp

  ld e,(ix+6) ; X
  ld d,(ix+4) ; Y
  pixelad
  setae
  or (hl)
  ld (hl),a

  pop ix
  ret
}

int abs(int x) {
  if (x<0) x=-x;
  return x;
}

void line(int x0, int y0, int x1, int y1) {
  int dx=abs(x1-x0);
  int sx=x0<x1?1:-1;
  int dy=-abs(y1-y0);
  int sy=y0<y1?1:-1;
```

The MSX that never was: a forgotten prototype

A journey into an alternative 1984, where a failed presentation changed the future of computing

by Tommaso Pitturru

For retrocomputing enthusiasts, the acronym MSX immediately evokes memories of a golden age of computing, that of the home personal computers of the 1980s. In a landscape dominated by giants such as Commodore with its C64 and Sinclair with the ZX Spectrum, the MSX standardâ€”adopted by the major consumer electronics manufacturers of the time, including Philips, Sony, Toshiba, Panasonic, Casio, etc.â€”was proposed as an alternative more geared towards the semi-professional sector, finding particular success in Japan, Europe, and South America. But at a crucial point in its evolution, things took a different turn. Today we tell you the story of an 'almost' 16-bit MSX: a prototype which, although based on two 8-bit processors, thanks to their ability to work in parallel, achieved such computing power and responsiveness that it rivalled the 16-bit machines that would be produced within the following year. A system so advanced for its time that it projected itself directly into the future, but condemned to oblivion by a decision that changed the course of home and personal computer history.

The story has been passed down to us by a direct source who has chosen to remain anonymous: we will call him Roberto, at the time Senior Systems Engineer for the European division of Digital Research, and witness to a confidential meeting held in London in the fall of 1984. The event was organized by Hitachi Europe with an ambitious goal: to present Kazuhiko Nishi, the "father" of MSX, returning from COMDEX, with a working prototype of a new generation MSX. Roberto was there to provide software support, given Digital Research's role in developing the operating system.

The heart of this revolutionary machine was not a single processor, namely the famous Z80 already used in MSX1 computers and others (Sinclair, Amstrad, and Tandy, for example), but an innovative dual Z80 processor architecture based on the high-performance 6MHz Z80B version. Instead of waiting for a new CPU, Hitachi engineers adopted a pragmatic solution already proven in other sectors such as arcade games (Galaga, Gyrus): one processor acted as the main CPU, while the second was configured as a powerful dedicated co-processor. Communication between the two was managed by a bus arbitration logic which, using the Z80's /BUSRQ and /BUSAK signals and a portion of shared RAM, allowed for effective parallel processing.

But the innovations did not stop at the CPU architecture. To ensure compatibility with the MSX standard, the video subsystem was entrusted to a not-yet-final version of the V9938 provided by Yamaha and later used for the MSX2. The chip interacted dynamically with the second Z80, which acted in effect as a dedicated graphics co-processor. This choice, in addition to being economically advantageous, drastically reduced programming complexity: developers could take advantage of the well-known and relatively simple architecture of the Z80 for graphics operations, without having to learn the instruction set of a custom co-processor: a radically different approach compared to contemporary systems, but at the same time easy to implement. For example, the second Z80 could perform complex graphics operations such as bit-block transfer (blitting), drawing geometric primitives, and manipulating bitmaps, while the V9938 simultaneously managed sprites and MSX compatibility modes, all without ever interrupting the main CPU.

The machine had 128 KB of RAM and 128 KB of dedicated video RAM, a respectable amount of memory for 1984.

The integration of the V9938 with a dedicated graphics co-processor unlocked a range of graphics modes that were significantly superior to the standards of the time, while maintaining backward compatibility. In addition to the standard MSX modes, the system offered:

Advanced Graphics Modes for GUI: resolutions up to 640Ã—400 with 16 colors from a palette of 512, or 384Ã—240 with 128 colors. Hardware support for fast window and icon movement made the GEM interface extremely responsive even at high resolutions, with a level of responsiveness that, at the time, even surpassed that of the Atari ST.

Advanced Sprite and Scrolling Management: Leveraging the native sprite management capabilities of the V9938 and the speed of the second Z80 in manipulating video memory, the prototype handled a greater number of hardware sprites simultaneously. In addition, display scrolling was smoother, without the "jerkiness" typical of many systems of the time.

Advanced Superimpose Functionality: It was possible to superimpose computer-generated graphics onto an external video signal with greater flexibility in the positioning and transparency of the graphic layers.
The audio subsystem was also upgraded to reflect the "Pro" status of the machine. A Yamaha YM2203 processor was chosen, which was also widely used in the arcade world. The choice was strategically perfect: it integrated a sound generator (PSG) fully compatible with the MSX1 chip, ensuring total audio backward compatibility, and added a 3-channel independent frequency modulation (FM) synthesizer. This hybrid architecture offered the best of both worlds, with a sound richness far superior to that available on the Atari ST.

The prototype was completed by an expensive and still uncommon, for the time, 3.5" floppy drive and a mouse. The operating system was a winning combination: Concurrent CP/M and GEM. During the demo, Nishi was shown the system running SuperCalc and MSX Basic in another window in multitasking mode. Thanks to the two processors working in parallel, the two applications ran with impressive fluidity, a stunning demonstration of the machine's capabilities. Suffice it to say that multitasking mode was not even available on the Apple Macintosh, which had recently come on the market.

According to our witness, despite the prototype's astonishing potential, the project was rejected at birth because of Nishi's perception of it. Although it was explained to him that the prototype was a proof-of-concept for a potential Pro version of the upcoming MSX2, to be positioned in the same market segment as the Apple Macintosh, Nishi remained skeptical about the complexity of the dual-processor architecture. It should be remembered that Nishi was a fervent advocate of standardization and simplicity in hardware design, convinced that the proliferation of proprietary and complex solutions would only fragment the market and confuse consumers. His vision for the MSX was that of an open and accessible platform, based on common and well-defined components. He considered the path from prototype to final product to be commercially risky, partly due to fears that costs would not be kept within acceptable limits.

Added to this was, perhaps, a previous distrust of Hitachi, which, despite being an MSX partner, did not seem particularly committed to advancing the standard. Not as much as Philips, Panasonic, or Sony, at least. But there was also another consideration, more subtle but no less influential. The use of Concurrent CP/M and GEM, products of Digital Research, placed the prototype in direct competition with the Microsoft ecosystem, with which Nishi had a well-established and strategic relationship (MSX-DOS, for example, was a version of Microsoft's MS-DOS, not CP/M). Nishi was well aware that Microsoft was developing its own graphical interface, Windows, which had already been announced in 1983. Although Windows would have had nothing to do with the MSX standard, adopting a solution based on the products of a direct competitor of Microsoft, however technically sound, could have created friction or been perceived as a "slap in the

face" to a key partner for the future of MSX. This strategic consideration, combined with technical reservations, contributed to his decision not to give the project the green light.

In addition to this, preparations for the MSX2 were already well underway and promised to significantly improve the native features of the standard.

So, despite the machine being particularly responsive and compatible, at least on paper, with all existing MSX software, Nishi categorically rejected the proposal.

We are thus catapulted back into our space-time continuum, where a prototype that never existed remains the dream of that young boy who took his first steps in the world of computing in those very years. He was different from the others, who, instead of being given the usual Commodore 64 or Spectrum, opted for those three still mysterious letters that made up the acronym MSX. That boy, the author of this article, now makes a living from computing and enjoys writing ucronic stories where he dreams of what might have been, imagining a future that we narrowly missed out on.

Visual Basic 6 on a web browser?

by Francesco Fiorentini

Those who have been following us for some time may remember an article I published on page 6 of issue 18 of RetroMagazine (at the time we still identified ourselves as RetroMagazine and not RetroMagazine World) entitled "Microsoft Visual Basic."

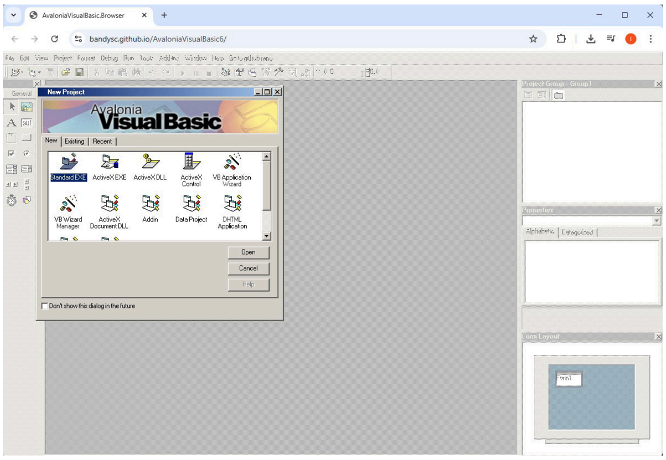
In that article, I wanted to pay tribute to the RAD (Rapid Application Development) suite par excellence, Visual Basic, which had the merit (or the flaw, according to some...) of opening the door to visual programming to anyone with a minimum of familiarity with code.

Visual Basic 6 was released by Microsoft in 1998 and remained an official commercial product until 2008. As I wrote in the article back in 2019, "However, the language is so popular that even today, the VB6 development team is committed to ensuring that it works on the latest operating system from Redmond: Windows 10."

These words could not have been more prophetic, given that in November 2024, coder Bartosz KorczyÅ„ski, alias Bandysc, released Avalonia Visual Basic 6.

In the words of its creator, Avalonia Visual Basic 6 "is nothing more than a reproduction of the classic Visual Basic 6 IDE and language in C# using Avalonia*. It is an amateur project with no commercial purpose. All rights to the Visual Basic name, icons, and graphics belong to Microsoft Corporation."
*Avalonia, on the other hand, is a cross-platform UI framework for dotnet that offers a flexible styling system and supports a wide range of platforms such as Windows, macOS, Linux, iOS, Android, and WebAssembly.

Obviously, this is not a complete version of Visual Basic, as code support is limited (...it doesn't say how much), but just seeing the screen that awaits us when we access the URL: https://bandysc.github.io/AvaloniaVisualBasic6/ anyone who has used Visual Basic 6 at least once, cannot help but feel a twinge of nostalgia.

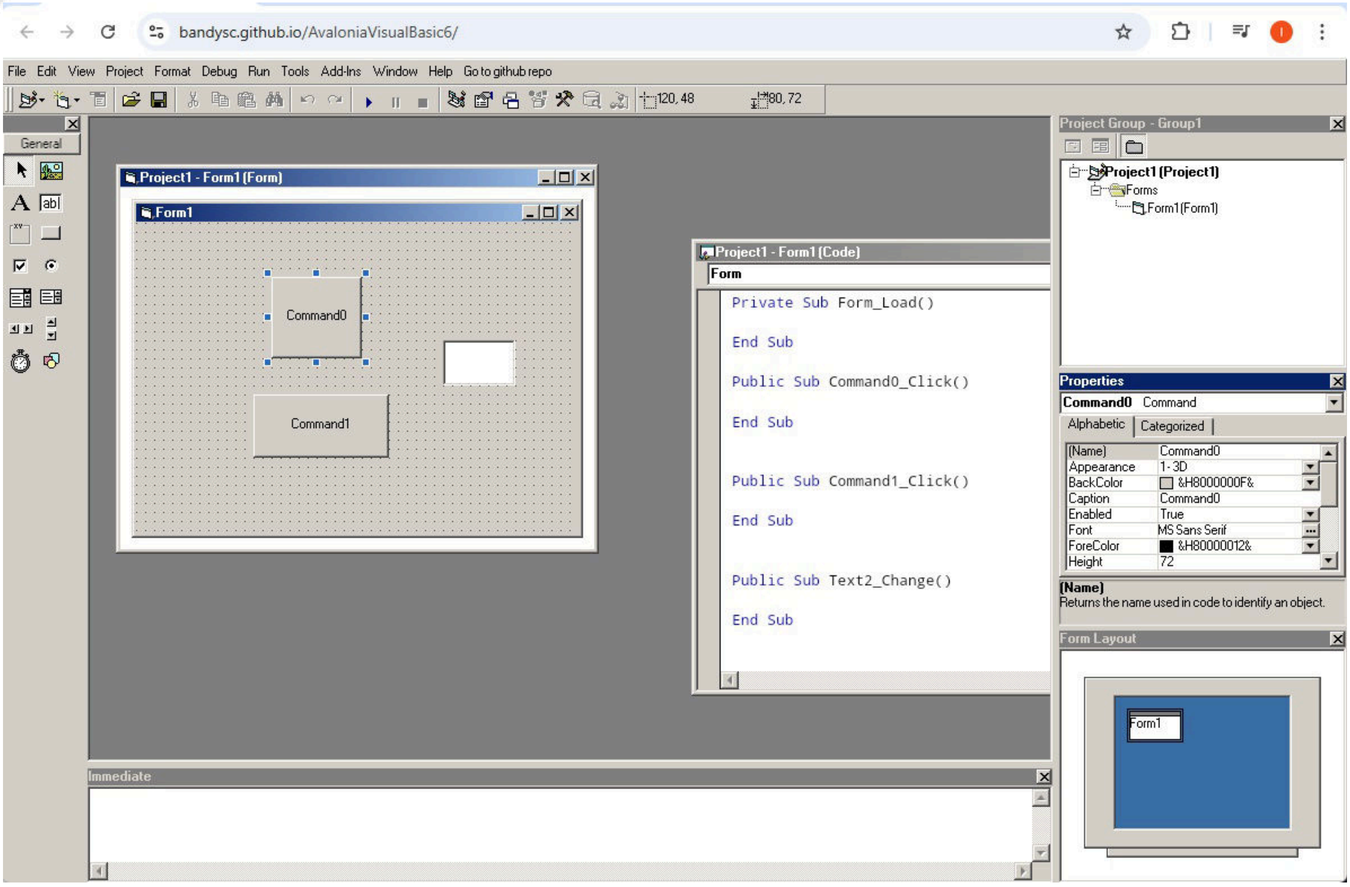


The Avalonia Visual Basic 6 interface is identical in every way to that of Visual Basic 6. In a word, incredible!
As we continue our exploration, everything becomes even more satisfying. Form1 is waiting for us inside the Project1 - Form1 (Form) window, ready to accept controls (only the default ones). Even the Properties window is identical to the original one but, if my memory serves me right, the list is a little shorter than usual (of course...).

So is it complete? Unfortunately not.
It is when adding controls to the Form that we notice the limitations of Avalonia Visual Basic 6. The events associated with objects are rather limited (and it couldn't be otherwise, given the amount of work behind a suite like VB6).

As can be partially seen in Figure 1, the events are limited to:
- Form - Load
- Timer - Timer
- Command - Click
- Text - Change
- Picture - Click

...and so on.



```
Private Sub Form_Load()

End Sub

Public Sub Command0_Click()

End Sub


Public Sub Command1_Click()

End Sub


Public Sub Text2_Change()

End Sub
```

(Name)	Command0
Appearance	1 - 3D
BackColor	&H8000000F&
Caption	Command0
Enabled	True
Font	MS Sans Serif
ForeColor	&H80000012&
Height	72

(Name)
Returns the name used in code to identify an object.

Fig. 1 - The Avalonia Visual Basic 6 graphical interface, pretty identical to Visual Basic 6

Obviously, Basic functions are also limited, making everything very pleasing to the eye but practically unusable for anything more than simple tests.

Oh, I almost forgot, as described in the project's GitHub repository, it is also possible to create a desktop version of Avalonia Visual Basic 6.

To do this, you need .NET 9.0, although you can modify Directory.Build.props to use .NET 8.0. To compile, simply run:

```
dotnet build AvaloniaVisualBasic.Desktop/AvaloniaVisualBasic.Desktop.csproj
```

If you encounter Antlr4 errors, it is likely that the Antlr4BuildTasks library failed to automatically download Java. Manually installing Java should resolve the issue.

Then run the commands:

```
dotnet publish AvaloniaVisualBasic.Desktop -f net9.0 -o bin/
dotnet publish AvaloniaVisualBasic.Standalone -f net9.0 -o bin/standalone/
```

to publish the IDE project (AvaloniaVisualBasic.Desktop) and the runtime (AvaloniaVisualBasic.Standalone).

Then running the command:

```
./bin/AvaloniaVisualBasic.Desktop
```

will launch the desktop version!

Enjoy this innovative tool to develop little applications in VB6 using a simple browser, maybe this project will evolve into something even more complex and powerful. It really winks an eye to nostalgia but it's great fun! Defintely it's worth a try.

Useful Links

Avalonia Visual Basic 6:
- https://bandysc.github.io/AvaloniaVisualBasic6/

AvaloniaVisualBasic6 project repository:
- https://github.com/BAndysc/AvaloniaVisualBasic6

Avalonia project repository::
- https://github.com/AvaloniaUI/Avalonia

.NET 9.0:
- https://dotnet.microsoft.com/en-us/download/dotnet/9.0

Rotating octahedron on the C64

From calculation to bitmap mode display

by Felice Nardella aka “Kimono”

In this article, I will try to explain how to represent a three-dimensional figure, such as an octahedron, on our Commodore 64 (abbreviated C64), simulating its rotation in space through simple geometric transformations and direct rendering in Bitmap mode.

I will start with the basics of three-dimensional rotations, then see how the vertices of the octahedron are projected onto the plane and finally, how they are drawn on the screen. The code is written in Assembly and handles rotation, perspective projection, and real-time visualization.

To simplify processing on the C64, the Assembly program does not directly calculate the coordinates of the rotated vertices. Instead of performing rotations and projections in real time on the C64 (which would require trigonometric calculations that are difficult to implement in 8 bits), all frames of the animation are precalculated and stored as static data. This approach is common in programming games and demos for 8-bit systems such as the C64.

For this reason, I wrote a Python program that performs these calculations separately and in advance, generating a series of coordinates ready to be inserted into the Assembly code. The C64 will only read the data in memory and draw the corresponding lines.

The animation has been made fluid through the use of double buffering: while one frame is displayed on the screen, the next is prepared in a second bitmap memory area. At the appropriate moment, the two buffers are swapped, thus avoiding the "flickering" effect.

The goal is not only to show a graphic effect, but also to offer an educational overview of how mathematics, computer science, and a passion for retrocomputers can come together in a creative way.

Preliminary Concepts on Rotations in the Plane

Rotation is a type of geometric transformation that rotates an object around a fixed point without changing its shape or size. We can imagine it as the movement of a clock hand, or as a ruler rotating around a drawing pin fixed to a sheet of paper.

In the Cartesian plane, both a single point and an entire figure, such as a triangle or a square, can rotate, changing position while remaining at the same distance from a fixed point, called the center of rotation, which often coincides with the origin of the axes, i.e., the point (0, 0).

In the Cartesian plane, the counterclockwise rotation of a point P(x, y) around the origin, by a positive angle Î± (in this article we will assume that the rotation is always counterclockwise), produces a new point P'(x', y') whose coordinates are calculated using the following formulas:

x^'=x•cosα-y•sinα

y^'=x•sinα+y•cosα

These equations are derived from linear algebra and can be easily obtained through simple trigonometric steps.

Let's now look at an example to understand how to apply them: imagine rotating a triangle in the Cartesian plane by an angle Î±. To do this, we will have to apply the rotation formulas to each vertex of the triangle.

Let's assume that the vertices of the triangle are A(1,2), B(4,1), and C(3,4) and that the angle Î± measures 90Â° (Ï€/2).

To perform this rotation, we will have to perform the following calculations, based on the above formulas:

1. Vertex A(1,2)

$x'=1\bullet\cos(90°)-2\bullet\sin(90°) = 1 \bullet 0 - 2 \bullet 1 = -2$

$y'=1\bullet\sin(90°)+2\bullet\cos(90°) = 1 \bullet 1 + 2 \bullet 0 = 1$

Therefore, the new vertex will have coordinates A'(-2,1)

2. Vertex B(4,1)

$x'=4\bullet\cos(90°)-1\bullet\sin(90°) = 4 \bullet 0 - 1 \bullet 1 = -1$

$y'=4\bullet\sin(90°)+1\bullet\cos(90°) = 4 \bullet 1 + 1 \bullet 0 = 4$

Therefore, the new vertex will have coordinates B'(-1,4)

3. Vertex C(3,4)

$x'=3\bullet\cos(90°)-4\bullet\sin(90°) = 3 \bullet 0 - 4 \bullet 1 = -4$

$y'=3\bullet\sin(90°)+4\bullet\cos(90°) = 3 \bullet 1 + 4 \bullet 0 = 3$

Therefore, the new vertex will have coordinates C'(-4,3)

Now we can represent the rotated triangle with the new vertices A'(x'_A, y'_A), B'(x'_B, y'_B), and C'(x'_C, y'_C).

Here is the graph of the original triangle with vertices A(1,2), B(4,1), C(3,4) and its counterclockwise rotation of 90°, with vertices:

$A' = (-2, 1)$

$B' = (-1, 4)$

$C' = (-4, 3)$

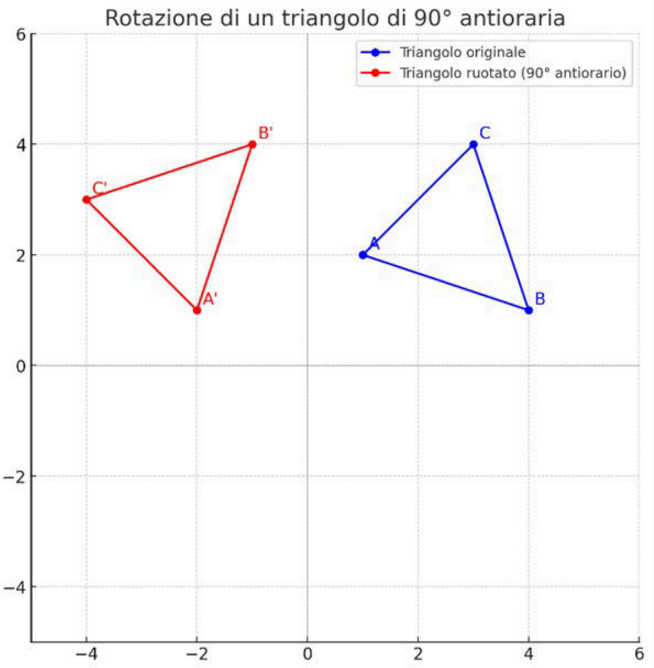


Fig.1 - Rotation of a triangle

Rotations in 3D space around a single axis

In the real world, figures are not only found on a plane: they occupy three-dimensional space, like a box, a chair, or a cube. Therefore, rotation also becomes three-dimensional: it no longer rotates around a point, but around a straight line, called the axis of rotation. Rotation in space, as in a plane, does not change the shape or size of the object, but it does change its position and orientation.

In three-dimensional space, rotations become more complex, as we can define separate rotations around each of the three main axes (X, Y, and Z). Each rotation involves only two of the three coordinates, while the axis around which they rotate remains unchanged.

Here is how rotations around individual axes behave:

Rotation of a point P(x,y,z) around the X axis by an angle α:

$x' = x$

$y' = y \cdot \cos(\alpha) - z \cdot \sin(\alpha)$

$z' = y \cdot \sin(\alpha) + z \cdot \cos(\alpha)$

As mentioned, the X axis does not change (in fact, $x' = x$) and the rotation takes place in the YZ plane, as if we were projecting the point onto that plane and rotating it byi α around the origin.

Rotation around the Y axis by an angle α:

$x' = x \cdot \cos(\alpha) + z \cdot \sin(\alpha)$

$y' = y$

$z' = -x \cdot \sin(\alpha) + z \cdot \cos(\alpha)$

Similarly, the Y axis does not change, while the rotation occurs in the XZ plane, as if we were projecting the point onto that plane and rotating it by an angle α.

Rotation around the Z axis by an angle α:

$x' = x \cdot \cos(\alpha) - y \cdot \sin(\alpha)$

$y' = x \cdot \sin(\alpha) + y \cdot \cos(\alpha)$

$z' = z$

In this case, the Z axis does not change and the rotation occurs in the XY plane, just as in the two-dimensional rotations seen above.

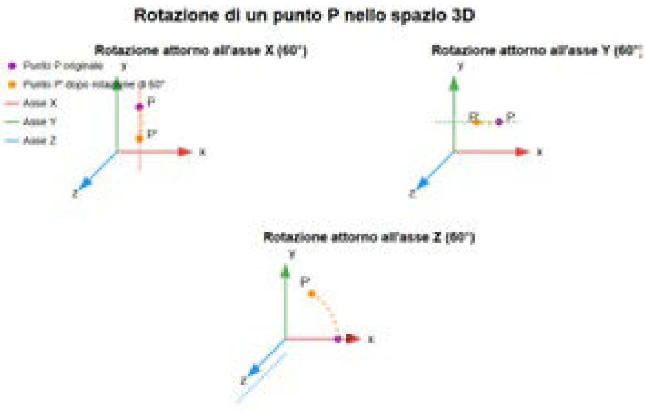

Rotazione di un punto P nello spazio 3D
Rotazione attorno all'asse X (60°)
Punto P originale
Punto P' dopo rotazione di 60°
Asse X
Asse Y
Asse Z
P
P'
x
y
z
Rotazione attorno all'asse Y (60°)
P'
P
x
y
z
Rotazione attorno all'asse Z (60°)
P'
x
y
z

Fig.2 - Rotation of a point P in space

Rotations in 3D space around combinations of axes

In a three-dimensional space "X-axis, Y-axis, and Z-axis" we can rotate an object around only one of these axes, but also around two axes simultaneously, or around all three. In cases of rotation around multiple axes, the overall rotation is obtained by combining the individual rotations around each axis, following a certain order.

There are several possibilities. Here are some examples:

- Rotation around the X-axis and around the Y-axis
- Rotation around the X-axis and around the Z-axis
- Rotation around the Y-axis and around the Z-axis
- Rotation around the X, Y, and Z axes

It should be noted that, in these cases, the order in which the rotations are performed is crucial, as the overall result depends on their sequence. Therefore, a rotation around X, then Y, then Z is not the same as performing a rotation around Z, then Y, then X. Every possible combination of rotations around X, Y, and Z leads to a different final result depending on the order in which they are applied.

In my case, I rotated the octahedron by applying two consecutive transformations to its points (the vertices), first on the X axis and then on the Y axis.

First rotation on the X axis: each point is "moved" as if it were rotating around the horizontal axis (the X axis). This changes the vertical position (y) and depth (z) of the point, leaving the lateral position (x) unchanged.

Then rotation on the Y axis: once the first rotation is complete, the new points are further rotated around the vertical axis (the Y axis). This operation changes the lateral position (x) and depth (z) of the point, leaving the height (y) unchanged.

Thanks to these two consecutive transformations, the octahedron appears to rotate in three dimensions, with independent effects on both axes.

Perspective Projection

At this point, we have a problem: the coordinates we have are in three-dimensional space, while the screen on which we want to display them is two-dimensional. To solve this problem, we use perspective projection.

Perspective projection is fundamental in computer graphics for transforming three-dimensional objects into a two-dimensional representation, simulating human visual perception.

Perspective projection is used to simulate the perception of depth, making objects appear smaller when they are further away.

For a point P(x, y, z) in 3D space, its projection P'(x', y') onto a projection plane located at a distance d from the observer is:

$x' = (x \cdot d)/(d+z)$

$y' = (y \cdot d)/(d+z)$

Where:

x' and y' are the coordinates projected onto the 2D plane

(x, y, z) are the coordinates of the point in 3D space

d is the distance between the observer and the projection plane

This formula produces the perspective effect in which:

objects closer to the observer (smaller z) appear larger

objects farther away (larger z) appear smaller

parallel lines converge toward vanishing points

The formula assumes that the observer is positioned at the origin O(0,0,0) and looks along the positive Z axis, with the projection plane perpendicular to the Z axis.

Here is an example of a perspective projection of two points P1 and P2 in the XY plane: (Fig.4).

Analysis of the Python Code for Octahedron Rotation

After this necessary introduction to geometric transformations, we can finally focus on programming.

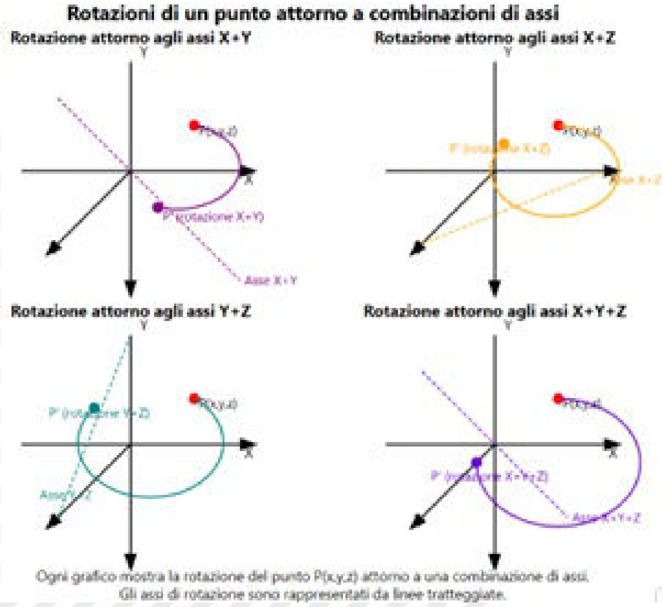

Rotazioni di un punto attorno a combinazioni di assi
Rotazione attorno agli assi X+Y
Rotazione attorno agli assi X+Z
Rotazione attorno agli assi Y+Z
Rotazione attorno agli assi X+Y+Z
Y
X
P(x,y,z)
P' (rotazione X+Y)
Asse X+Y
P' (rotazione X+Z)
Asse X+Z
P' (rotazione Y+Z)
Asse Y+Z
P' (rotazione X+Y+Z)
Asse X+Y+Z
Ogni grafico mostra la rotazione del punto P(x,y,z) attorno a una combinazione di assi.
Gli assi di rotazione sono rappresentati da linee tratteggiate.

Fig.3 - Rotations of a point around combinations of axes

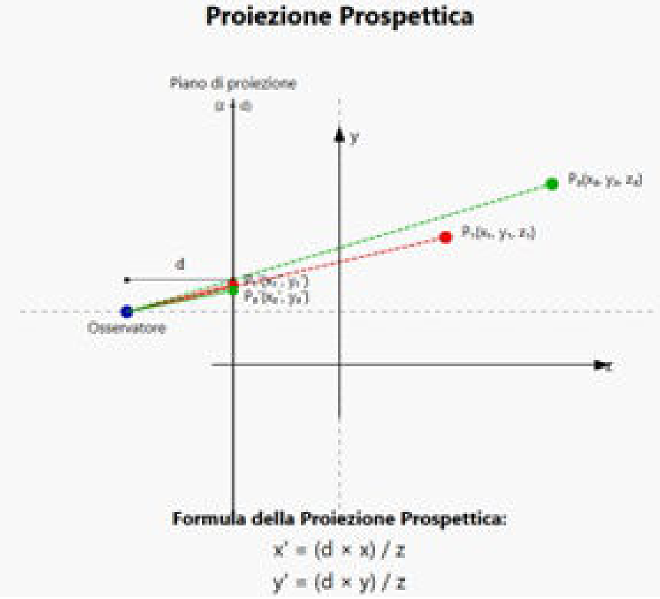


Fig.4 - Perspective projection of two points P1 and P2 in the plane

To calculate the coordinates of the octahedron's vertices, I used a code I wrote in Python.

What is an octahedron?

An octahedron is a regular polyhedron formed by 6 vertices, 12 edges, and 8 equilateral triangular faces. It is one of the five Platonic solids (along with the tetrahedron, cube, dodecahedron, and icosahedron) and visually, it can be imagined as two square-based pyramids joined at their bases.

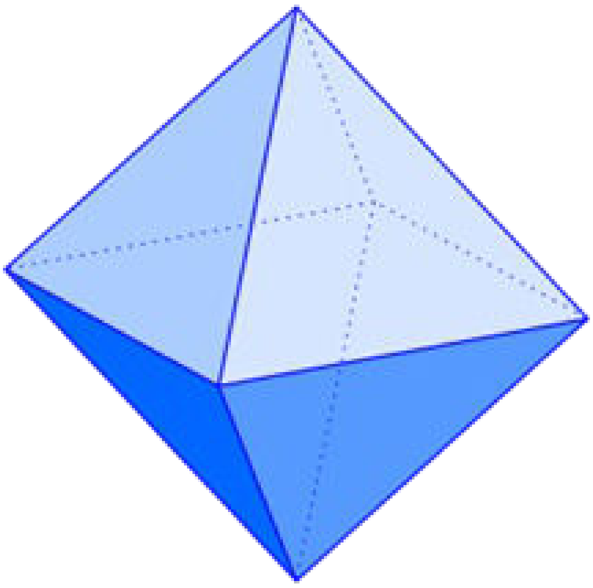


Fig.5 - Example of an octahedron

Description of the Python code

The code calculates the coordinates of the vertices of an octahedron, simulates its rotation in 3D space, and generates precalculated data to be used directly in the Assembly program for our C64. Let's analyze it in detail:

Definition of the geometry of the octahedron

```
def generate_octahedron_vertices():
  coords = [
    [ 1,  0,  0],  # 0
    [-1,  0,  0],  # 1
    [ 0,  1,  0],  # 2
    [ 0, -1,  0],  # 3
    [ 0,  0,  1],  # 4
    [ 0,  0, -1]   # 5
  ]
  # Scale for display
  scale = 50
  return [[x * scale, y * scale, z * scale] for x, y, z in coords]
```

First, we need to define the vertices: as mentioned, the octahedron is a polyhedron with 6 vertices. I have therefore defined its vertices, placing them at a unit distance from the center, along each coordinate axis. These represent the so-called standard coordinates for the octahedron:

[1, 0, 0] # 0

This vertex is located along the positive X-axis, at a unit distance (1) from the origin. The other Y and Z axes have a value of 0.

[-1, 0, 0] # 1

This vertex is located along the negative X-axis, again at a unit distance from the origin. Here too, Y and Z are 0.

[0, 1, 0] # 2

This vertex is located along the positive Y axis, at a unit distance from the origin. The X and Z axes have a value of 0.

[0, -1, 0] # 3

This vertex is located along the negative Y axis, at a unit distance from the origin. Again, X and Z are 0.

[0, 0, 1] #4

This vertex is located along the positive Z-axis, at a unit distance from the origin. The X and Y axes are 0.

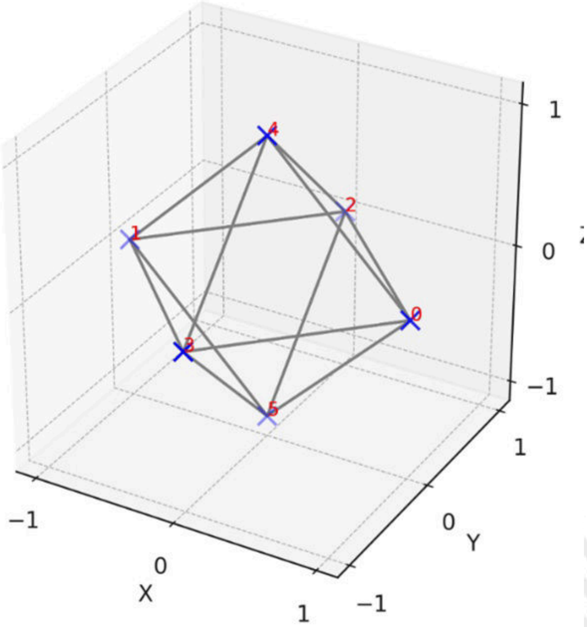


Fig.6 - Standard coordinates for the octahedron and their connections

Next, these are scaled for better display on the C64 screen, so each coordinate is multiplied by a scale (in this case 50) to make it visible. For example, the vertex [1, 0, 0] becomes [50, 0, 0].

Definition of Connections

```
def generate_octahedron_lines():
  return [
        [0, 2], [0, 3], [0, 4], [0, 5],
        [1, 2], [1, 3], [1, 4], [1, 5],
        [2, 4], [2, 5], [3, 4], [3, 5]
    ]
```

This function defines which vertices are connected to each other to form the edges of the octahedron (see Figure 6). In total, the octahedron has 12 edges, each represented by a pair of indices that refer to the vertices defined above.

3D rotations

```
def rotate_x(point, angle):
    """ Rotates a point around the X axis """
    y = point[1] * math.cos(angle) - point[2] * math.sin(angle)
    z = point[1] * math.sin(angle) + point[2] * math.cos(angle)
    return [point[0], y, z]

def rotate_y(point, angle):
    """ Rotate a point around the Y axis """
    x = point[0] * math.cos(angle) + point[2] * math.sin(angle)
        z = -point[0] * math.sin(angle) + point[2] *
math.cos(angle)
    return [x, point[1], z]
```

These functions calculate the rotation of each point around the X and Y axes:

rotate_x: Rotates the point around the X axis using the trigonometric formulas described above.

rotate_y: Similarly, rotates the point around the Y axis.

This rotation allows you to simulate the animation of the octahedron rotating in space. Each function takes a 3D point and a rotation angle, and returns the point rotated according to the specified angle. Note that, as already explained, during rotation around the X axis, the x coordinate remains unchanged, just as, in the case of rotation around the Y axis, the y coordinate remains unchanged.

3D projection in 2D

```
def project_point(point):
    """ Simple perspective projection """
    scale = 200 / (200 + point[2])
    x = point[0] * scale + CENTER_X
    y = point[1] * scale + CENTER_Y
    return [int(x), int(y)]
```

The project_point() function accepts a single parameter (point), which is a list containing the three values [x, y, z]. This function implements a simple perspective projection, which simulates the effect of depth, by applying the formulas seen above. This formula calculates a scale factor based on the Z coordinate of the point (point[2]). The most distant points (with greater Z) will appear smaller on the screen. The values 200 in the formula represent the virtual focal distance (d).

The coordinates are then translated to center them on the screen. In the C64's bitmap mode, the resolution is 320 pixels horizontally (X axis) and 200 pixels vertically (Y axis), so the centering values are CENTER_X = 160 and CENTER_Y = 100.

Finally, the calculated X and Y coordinates are converted to integers (needed to draw pixels on the screen) and returned as a list [x, y].

Frame Generation

```
def generate_frames(num_frames=15):
    frames = []
    vertices = generate_octahedron_vertices()
    lines = generate_octahedron_lines()

    for frame in range(num_frames):
    # Calculate two different rotation angles
    angle_x = (frame * 2 * math.pi) / num_frames
     angle_y = (frame * 2 * math.pi) / (num_frames * 2)  #
Rotate more slowly on Y

      # Rotate first on X and then on Y
      rotated_vertices = [rotate_x(v, angle_x) for v in vertices]
          rotated_vertices = [rotate_y(v, angle_y) for v in
rotated_vertices]

      # Project the rotated vertices
          projected_vertices = [project_point(v) for v in
rotated_vertices]

      # Generate the lines for this frame
      frame_lines = []
      for line in lines:
        start = projected_vertices[line[0]]
        end = projected_vertices[line[1]]
            frame_lines.append([start[0], start[1], end[0],
end[1]])

      frames.append(frame_lines)
return frames
```

The generate_frames() function is the heart of this project, as it generates a sequence of frames in which the octahedron rotates in space. As you can see, I chose to generate 15 frames for each complete rotation. Of course, the number of frames can be increased or decreased as desired, but if you choose to increase it too much, it is essential to consider the memory limitations of the C64, which will have to store all the calculated coordinates. For this reason, 15 frames seemed to me to be a good compromise between smooth animation and memory space optimization.
The code begins with the creation of three fundamental elements:

- A list called frames, which will collect the data for each frame.
- The vertices, i.e., the three-dimensional coordinates of the six vertices of the octahedron.
- The lines, i.e., the connections between the vertices, which define its structure.

For each frame, the two rotation angles are calculated:

- angle_x - determines the angle of rotation around the X axis. This value ranges from 0 to 2Ï€, ensuring a complete rotation in 15 frames.
- angle_y - is similar, but moves more slowly because it is divided by twice the number of frames (num_frames * 2). This slows down the rotation around the Y axis.

The angle of rotation between one frame and the next is calculated by dividing a complete turn (2Ï€ radians, or 360Â°) by the number of frames:

- For the X axis: angle_x = $2\pi/15 \approx 0.42$ radians = 24° per frame
- For the Y axis: angle_y = $2\pi/(15\times2) = \pi/15 \approx 0.21$ radians = 12° per frame

This means that while the octahedron rotates completely around the X axis in a certain number of steps, its rotation around the Y axis occurs more slowly.
Now that we have the angles, it's time to rotate the solid by rotating each vertex around the X axis, changing the depth (z) and height (y). Next, the new vertices are rotated around the Y axis, affecting the width (x) and depth (z). Thanks to these sequential transformations, the octahedron rotates, moving in space.
After rotation, each vertex is converted from three-dimensional to two-dimensional coordinates so that it can be drawn on the screen using the project_point() function, which handles this step by simulating perspective, i.e., the effect whereby distant objects appear smaller.
Now that we have the projected points, we can connect the vertices to obtain a visible drawing. Each pair of vertices is transformed into a line, creating the outline of the rotated octahedron.
Once completed, the entire frame is stored in the frames list, ready to be used in the animation.

Assembly Code Generation

```
frames = generate_frames()
print("; Precalculated coordinates for the octahedron frames")
print("\nFRAMES")
for frame_num, frame in enumerate(frames):
   print(f"\n; Frame {frame_num}")
   for line in frame:
   print(f"    .byte {line[0]}, {line[1]}, {line[2]}, {line[3]}")
```

The final part of the code prints the precalculated data in a format readable by Assembly for the C64. Each line segment is converted into four bytes representing the initial and final coordinates of the segment on the screen. Then, for each frame, the 2D coordinates (x1, y1, x2, y2) of each line are listed.

Running the code

Once you have written the program in a text file using any editor (such as Notepad, Visual Studio Code, or Sublime Text), you need to save it with the .py extension, which identifies Python files. At this point, to run the script, simply open a terminal or command prompt, navigate to the folder where the file was saved, and run the command:

```
python octahedron_rotation.py
```

I personally used Python version 3.13.1, but the program should work without any problems with any version from 3.6 onwards.
Figure 7 shows part of the output generated by the

program, which highlights the correct functioning of the written code.

To use the data produced, simply select the text displayed in the output, copy it, and paste it into the Assembly code.

Explanation of the Assembly Code for C64

In this section, I will analyze step-by-step the Assembly code for C64 that displays and rotates the octahedron, using the double buffering technique in bitmap mode.

Code Structure and Organization

The program implements a set of techniques that include:

- High-resolution bitmap mode - 320x200 monochrome pixels
- Double buffering - a technique that uses two memory areas for smooth animation
- Bresenham's algorithm - for drawing lines
- Precalculated frames - a sequence of predetermined coordinates for animation

The program is organized into several sections:

- Definition of constants and variables
- Initial setup
- Initialization of the double buffer
- Start of the animation cycle
- Precalculated frame data

This approach makes the code much more readable.

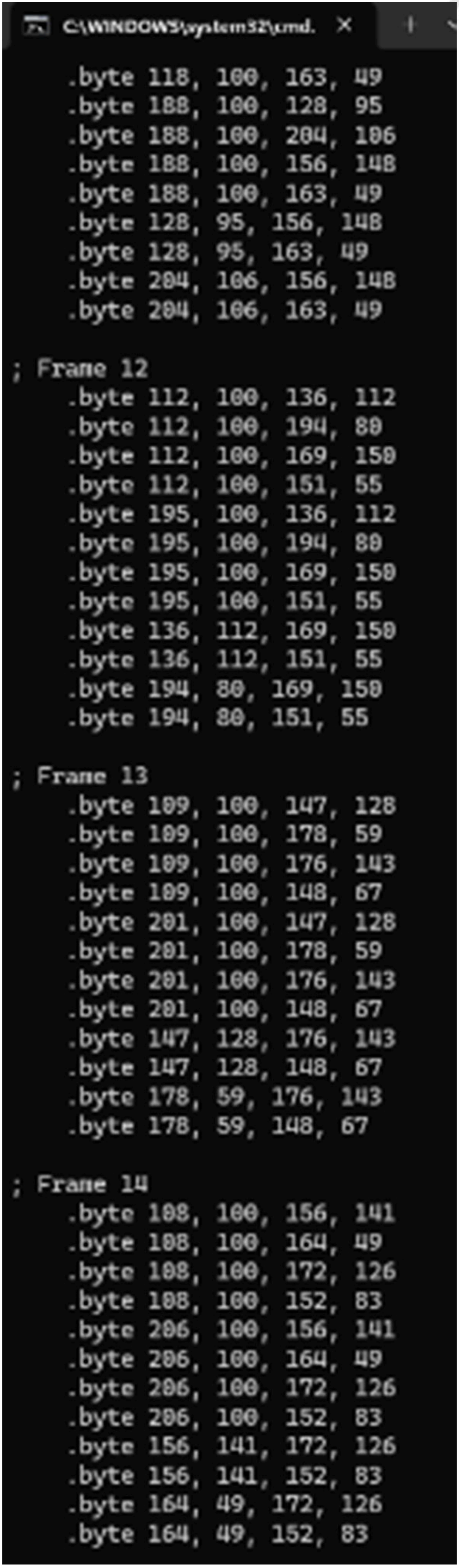
Definition of constants and variables

```
VICII = $d000 ; Base address of the VIC-II video chip
VICII1 = VICII + $11 ; VIC-II control register (bitmap mode)
VICIIA = VICII + $18 ; Graphics memory address register
VICIIB = VICII + $20 ; Border color register
VICIIS = VICII + $21 ; Background color register
BITMAP1 = $2000 ; First bitmap area (8KB)
BITMAP2 = $6000 ; Second bitmap area (8KB)
```

These definitions set the addresses of the VIC-II video chip and the memory areas where the two bitmaps for double buffering will be stored.

Main routine

```
START
  ldx #11
  stx VICIIB     ; Dark gray border
  jsr HIRES      ; Switch to hi res mode
```



```
    .byte 118, 100, 163, 49
    .byte 188, 100, 128, 95
    .byte 188, 100, 204, 106
    .byte 188, 100, 156, 148
    .byte 188, 100, 163, 49
    .byte 128, 95, 156, 148
    .byte 128, 95, 163, 49
    .byte 204, 106, 156, 148
    .byte 204, 106, 163, 49

; Frame 12
    .byte 112, 100, 136, 112
    .byte 112, 100, 194, 80
    .byte 112, 100, 169, 150
    .byte 112, 100, 151, 55
    .byte 195, 100, 136, 112
    .byte 195, 100, 194, 80
    .byte 195, 100, 169, 150
    .byte 195, 100, 151, 55
    .byte 136, 112, 169, 150
    .byte 136, 112, 151, 55
    .byte 194, 80, 169, 150
    .byte 194, 80, 151, 55

; Frame 13
    .byte 109, 100, 147, 128
    .byte 109, 100, 178, 59
    .byte 109, 100, 176, 143
    .byte 109, 100, 148, 67
    .byte 201, 100, 147, 128
    .byte 201, 100, 178, 59
    .byte 201, 100, 176, 143
    .byte 201, 100, 148, 67
    .byte 147, 128, 176, 143
    .byte 147, 128, 148, 67
    .byte 178, 59, 176, 143
    .byte 178, 59, 148, 67

; Frame 14
    .byte 108, 100, 156, 141
    .byte 108, 100, 164, 49
    .byte 108, 100, 172, 126
    .byte 108, 100, 152, 83
    .byte 206, 100, 156, 141
    .byte 206, 100, 164, 49
    .byte 206, 100, 172, 126
    .byte 206, 100, 152, 83
    .byte 156, 141, 172, 126
    .byte 156, 141, 152, 83
    .byte 164, 49, 172, 126
    .byte 164, 49, 152, 83
```

Fig.7 - Part of the output generated by the Python program

```
   jsr GRRAM       ; Set the position of the Bitmap
   jsr CLEARCO1    ; Clear and set Color RAM 1
   jsr CLEARCO2    ; Clear and set Color RAM 2
   jsr INIT_DOUBLE_BUFFER   ; Initialize double buffer
   jsr DRAW        ; Draw and rotate the polyhedron
   rts
```

The main routine:

- Sets the border color
- Activates high-resolution bitmap mode
- Initializes the memory areas for colors and graphics
- Configures the double buffering system
- Starts the octahedron drawing cycle

Explanation of function calls in Assembly code

Here is an explanation of each function call (subroutine) in the program:

HIRES:

Activate high-resolution graphics mode

```
HIRES
   lda VICII1
   ora #32        ; Set bit 5: bitmap mode active
   sta VICII1
   rts
```

The HIRES routine activates the C64's high-resolution graphics mode (bitmap) by modifying the VICII1 register (address $D011) and setting bit 5 (32 = %0010 0000) to switch from text mode to bitmap mode, which offers a resolution of 320x200 monochrome pixels.

GRRAM: Setting the graphics memory location

```
GRRAM
   lda VICIIA
   ora #$08
   sta VICIIA
   sta curpage
   rts
```

This routine configures the bitmap memory location in the VIC-II graphics chip. It modifies the VICIIA register (address $D018) by setting bit 3 to indicate that the bitmap starts at address $2000. It also stores this configuration in the curpage variable for future use.

CLEARCO1: Initialization of the first color area

```
CLEARCO1
   lda #208
   ldx #0
   stx $fa
   ldx #4        ; Sets the high byte of the address in $FB to 4 (address $0400)
   stx $fb
   ldy #0
cloop1
   sta ($fa),y
   iny
   bne cloop1
   inc $fb
   ldx $fb
   cpx #8        ; Checks if it has reached page $0800
   bne cloop1
   rts
```

This routine clears and initializes the first area of Color RAM (address $0400-$07EF, although here it actually goes up to $0800). It writes the value 208 to all bytes in this area, thus setting the color of the pixels and background in bitmap mode. In the C64 bitmap mode, each 8x8 pixel block can have a foreground color (determined by the Color RAM) and a background color. The value 208 is derived from the formula:

foreground color x 16 + background color

In our case, foreground color = 13 (light green) and background color = 0 (black)

CLEARCO2: Initialization of the second color area

Similar to CLEARCO1, but operates on the second area of Color RAM (address $4400-$47EF). This second area is necessary for double buffering, allowing separate color settings for each of the two bitmap pages.

INIT_DOUBLE_BUFFER: Preparing the double buffering system

```
INIT_DOUBLE_BUFFER
   ; Set the initial switch flag
   lda #1
   sta switch_flag

   ; Configure bank 0 (where BITMAP1 is located)
   lda $dd00
   and #252  ; Clear the 2 bank bits
   ora #3    ; Bank 0 ($2000-$3FFF)
   sta $dd00
    ; Clear both bitmaps
   jsr CLEAR_BITMAP1
   jsr CLEAR_BITMAP2
  rts
```

This routine initializes the double buffering system.

- The double buffering technique consists of:
- Having two complete memory buffers (BITMAP1 at $2000 and BITMAP2 at $6000)
- While one buffer is displayed, the other is updated with the next frame
- Once the drawing is complete, wait for the end of the scan (rasterline 0)
- Switch instantly between the two buffers
- Start drawing the next frame in the other buffer

Double buffering therefore draws a frame off-screen and then quickly swaps it with the visible one, avoiding the so-called flickering.

The "VIC Bank Select"

The VIC-II video chip cannot see all the RAM at once, but only one 16 KB bank at a time. There are 4 possible banks in the 64 KB RAM. The VIC-II reads graphic data (bitmaps, characters, sprites, colors) only from the active bank.

The $DD00 (56576) register of the C64 (which belongs to the CIA #2 chip) contains, in its bits 0 and 1, the values that control which 16KB bank is visible to the VIC-II. The possible values are:

%11 (3): Bank 0 ($0000-$3FFF)
%10 (2): Bank 1 ($4000-$7FFF)
%01 (1): Bank 2 ($8000–$BFFF)
%00 (0): Bank 3 ($C000–$FFFF)

Note that the values are inverted: the higher the value in the register, the lower the memory address.

Then we set the bits to ensure that the RAM needed for the bitmaps is active and accessible.

In the octahedron program, bank switching is used in the FLIP_PAGES routine:

```
FLIP_PAGES
   lda switch_flag
   eor #1          ; Invert the flag
   sta switch_flag
   beq USE_BITMAP1

   ; Switch to BITMAP2 (bank 1)
   lda $dd00
   and #252       ; Clear the 2 bank bits (leaves the other bits unchanged)
   ora #2         ; Bank 1 ($4000-$7FFF)
   sta $dd00

   ; Set the pointer for drawing on BITMAP1
   lda #>BITMAP1
   sta bmpage
   jmp DONE_FLIP

USE_BITMAP1
   ; Switch to BITMAP1 (bank 0)
   lda $dd00
   and #252
   ora #3         ; Bank 0 ($0000-$3FFF)
   sta $dd00

   ; Set the pointer for drawing to BITMAP2
   lda #>BITMAP2
   sta bmpage
```

This routine flips the switch flag (0 or 1):

If the flag is 0, it configures the VIC-II to display BITMAP1 (in bank 0) and sets the drawing to BITMAP2

If the flag is 1, it configures the VIC-II to display BITMAP2 (in bank 1) and sets the drawing to BITMAP1

The code modifies register $DD00 to change the bank visible to the VIC-II. The AND #252 (%1111 1100) operation clears bits 0 and 1 of the bank, leaving the other bits of the register unchanged (used for other functions), while the XOR #2 or XOR #3 operation sets the bank bits to the desired values:

ORA #2 selects bank 1 ($4000-$7FFF)
ORA #3 selects bank 0 ($0000-$3FFF)

Thus, while the VIC-II displays one bitmap page, the program draws on the other and then switches between the two.

DRAW:Start of the animation cycle

```
DRAW
```

```
    ; Initialize the pointer to the first frame
    lda #<COORD_TABLE
    sta FRAMEPTR
    lda #>COORD_TABLE
    sta FRAMEPTR+1

    ; Initialize the frame counter
    lda #0
    sta FRAMECNT

    ; Clear the page that is not currently displayed
    jsr CLEAR_BITMAP1
    ; Set the initial drawing buffer
    lda #>BITMAP1   ; Start drawing on the first page
    sta bmpage
```

After preparing the graphics mode and the two buffers, the program calls DRAW.

This routine:

- Initializes the pointers to the coordinate table (COORD_TABLE)
- Sets the frame counter (FRAMECNT) to 0
- Clears the first bitmap
- Sets the first bitmap page as the initial drawing buffer
- Starts the animation loop by calling the ANIMATE routine (implicitly)

CLEAR_CURRENT_BITMAP: Clearing the active drawing area

```
CLEAR_CURRENT_BITMAP
     ; Use the value in bmpage to determine which bitmap to clear
    lda bmpage
    cmp #>BITMAP1
    beq CLEAR_BITMAP1
    jmp CLEAR_BITMAP2
```

This routine clears the bitmap page that will be used to draw the next frame. It checks the value of the bmpage variable to determine which bitmap to clear (BITMAP1 or BITMAP2) and calls the appropriate routine.

DRAWFRAME: Rendering a single frame of the octahedron

```
DRAWFRAME
    ldy #0
    ldx #0
draw_loop
    ; Read coordinates x1,y1
    lda (FRAMEPTR),y
    sta x1
    iny
    lda (FRAMEPTR),y
    sta y1
    iny
    ; Read coordinates x2,y2
    lda (FRAMEPTR),y
    sta x2
    iny
    lda (FRAMEPTR),y
    sta y2
    iny
    stx store+1     ; Save X
    tya
    pha             ; Save Y
    jsr LINE        ; Draw the line
    pla
    tay             ; Retrieve Y
    ldx store+1     ; Retrieve X
    inx
    cpx #12         ; 12 lines for the octahedron
    bne draw_loop
    rts
```

This routine draws a single frame of the octahedron:

- Reads 12 pairs of coordinates (corresponding to the 12 edges) from the precalculated table
- For each pair of points, calls the LINE routine to draw a line between them
- The result is the display of the octahedron in a specific rotation position

ANIMATE: Frame-by-frame animation loop

```
ANIMATE
      jsr CLEAR_CURRENT_BITMAP   ; Clears the current bitmap before drawing on it
     jsr DRAWFRAME    ; Draws the current frame on the bmpage page

    lda FRAMEPTR
    sta store+2

    ; Wait for the end of the frame (rasterline 0)
    lda #255
WAIT_RASTER
```

```
    cmp VICII+$12   ; Compare with the rasterline register
    bne WAIT_RASTER

    ; Change pages
    jsr FLIP_PAGES

    ; Move to the next frame
    lda store+2
    clc
    adc #48        ; Each frame occupies 48 bytes
    sta FRAMEPTR
    bcc _no_carry
    inc FRAMEPTR+1
_no_carry
 ; Increment the frame counter
    inc FRAMECNT
    lda FRAMECNT
    cmp #15
    bne ANIMATE

    lda #0
    sta FRAMECNT

    lda #<COORD_TABLE
    sta FRAMEPTR
    lda #>COORD_TABLE
    sta FRAMEPTR+1
    jmp ANIMATE
```

The ANIMATE routine is responsible for advancing the animation on the screen, managing the sequence of frames smoothly.

When it starts, the first thing it does is clear the bitmap page on which it is about to draw. This is essential: it prevents traces of the previous drawing from remaining. As soon as the clearing is finished, it calls a second routine, DRAWFRAME, which draws the new frame: then, using the coordinates of the current frame, the object is drawn on the bitmap page.

At this point, the program must ensure that the drawing is displayed at the right time, avoiding interference with the work of the VIC-II.

For this reason, the routine enters a short active wait: it continuously checks the value of the rasterline (i.e., which line of the screen the VIC-II is drawing) and waits for it to reach 255, which is practically the end of the screen. Only when the raster has reached that point can it proceed without the risk of video artifacts.

Once the end of the video drawing has been reached, a page flip is performed: this means that the program swaps the active video page.

After displaying the new frame, the program must prepare for the next one. It then updates the frame data pointer (FRAMEPTR), moving forward 48 bytes, as in this case, each frame occupies exactly that amount of memory. If the increment operation results in a carry, the high byte of the pointer is also incremented.

The routine then increments a frame counter, called FRAMECNT, which keeps track of how many frames have already been displayed in the current sequence.

If it has not yet reached 15, the program simply starts over and continues the animation by drawing the next frame.

If, on the other hand, 15 frames have been displayed, the program understands that it has reached the end of the sequence. At that point, it resets the frame counter (FRAMECNT) and returns the FRAMEPTR pointer to the beginning of the data table (COORD_TABLE). Finally, it starts again with the first image.

This whole mechanism repeats itself indefinitely, creating a smooth animation loop, synchronized with the screen refresh.

Drawing a line: the Bresenham algorithm

The LINE and SUB_LINE routines implement the Bresenham algorithm to draw a line between two points (x1, y1) and (x2, y2), calculating the optimal trajectory between the two points.

This line drawing algorithm was developed by Jack E. Bresenham in 1962 while working at IBM.

The importance of this algorithm lies in its computational efficiency: it uses only integer arithmetic operations (additions, subtractions, and shifts), avoiding multiplications, divisions, and floating-point operations.

For this reason, it has become fundamental in computational graphics, especially on systems with limited resources such as the microcomputers of the 1980s.

Principle of the algorithm

For a line going from point (x0, y0) to point (x1, y1), the algorithm must determine which pixels to turn on to best approximate this line.

The key idea is:

- Determine which axis has the greater distance (horizontal or vertical)
- Advance one unit at a time along the dominant axis
- Decide when to advance along the other axis as well

So, the algorithm decides when to take a horizontal step, when to take a vertical step, and when to take both, in order to stay as close as possible to the imaginary line.

The algorithm maintains an "accumulated error" that helps decide when to take a diagonal step instead of a pure horizontal or vertical step.

Here is how the algorithm is structured, in the case where the slope (dy / dx) is between 0 and 1 and the increment occurs along the X axis:

1. Enter the coordinates of the two ends of the line: ($x1$, $y1$) and ($x2$, $x2$).

Set the first pixel to ($x1$, $y1$).

2. Calculate the differences:

$dx = (x2 - x1)$ e $dy = (y2 - y1)$.

3. Calculate the error value:

$E = 2dy - dx$.

4. For each step along the X axis, from $x = x1$ to $x2 - 1$:

- If $E < 0$:

  the next pixel is ($x + 1$, y)

  update the error: $E = E + 2dy$
- If $E \geq 0$:

  the next pixel is ($x + 1$, $y + 1$)

  update the error: $E = E + 2dy - 2dx$

5. Repeat step 4 until x reaches x2.

SUB_LINE: Calculation of initial parameters

```
SUB_LINE
    ; Check if x1 > x2 and, if so, swap the points
    lda x1
    cmp x2
    bcc no_swap_x   ; If x1 < x2, no need to swap

; [code to swap points]

   no_swap_x
    ; Calculate dx = abs(x2 - x1)
    lda x2
    sec
    sbc x1
    bpl store_dx    ; If positive, skip
    eor #$FF        ; Otherwise, negate (complement to 1)
    clc
    adc #1          ; Complement to 2 to obtain abs()
store_dx
    sta dx

; [similar code to calculate dy = abs(y2 - y1)]

    ; Determine x direction (sx)
    lda x1
    cmp x2
    bcc x_ascending
    lda #$FF        ; sx = -1 (x1 > x2)
    jmp store_sx
x_ascending
    lda #$01        ; sx = 1 (x1 < x2)
store_sx
    sta sx

    ; [similar code to determine sy]

    ; Set the initial values of x and y
    lda x1
    sta xL
    lda #0
    sta xH
    lda y1
    sta yy
    rts
```

This is the part that takes care of preparing everything necessary before starting to draw.

1. Always sort points from left to right

The SUB_LINE subroutine starts by checking which of the two points is further to the left. If x1 > x2, i.e. if the starting point is further to the right than the end point, then the two points are swapped. This is for the sake of simplicity: drawing from left to right avoids a lot of complications in comparisons and loops.

2. Calculate dx and dy (horizontal and vertical distances)

After arranging the order of the points, two fundamental values are calculated:

- dx: how far to go horizontally
- dy: how far to go vertically

But be careful: we are not interested in whether the difference is negative or positive; therefore, we always take the absolute value, i.e., the distance "forward," regardless of the direction.

3. Establish the direction of movement

Now we need to understand which direction to go:

- If the second point is further to the right, then sx = 1
- If it is further to the left, then sx = -1 (which becomes $FF in binary format)

The same applies to sy, i.e., the vertical direction.

4. Set the starting point

Finally, set the starting coordinates of the line: xL (lower part of the X, since it is 16 bits), xH (the upper part, which is set to zero for simplicity), and yy, which is the initial Y coordinate.

At this point, we are ready to actually start drawing the line.

LINE: Bresenham's main logic

```
LINE
  jsr SUB_LINE

  ; Calculate err = (dx > dy ? dx : dy) / 2
  lda dx
  cmp dy
  bcc dy_greater
  lsr         ; Divide dx by 2
  sta err
  jmp check_slopes
dy_greater
  lda dy
  lsr         ; Divide dy by 2
  sta err

check_slopes
  ; Decide which loop routine to use
  lda dx
  cmp dy
  bcs dx_ge_dy   ; If dx >= dy, use the x-dominant routine
```

This part contains the main logic of the design, which actually executes the steps one at a time.

Which is more "dominant": dx or dy?

First, dx and dy are compared. This is to understand whether the line is "more horizontal" or "more vertical."

- If the line is wider than it is tall (dx >= dy), a loop will be used where a horizontal step is always taken and a decision is made whether to also take a vertical step.
- If, on the other hand, it is taller than it is wide (dy > dx), the opposite is done: a vertical step is always taken, and a decision is made as to whether to also take a horizontal step.

This distinction is crucial in order to keep the line as 'straight' as possible, without making it look broken or stepped.

The error: the heart of the algorithm

The algorithm keeps track of a value called err, which basically measures how much we are "derailing" from the ideal line.

At the beginning, err is set to half the dominant value (either dx/2 or dy/2). This value will then be updated at each step.

Loop for dominant lines in x (x_loop)

```
x_loop
  jsr PLOT      ; Draw the point

  ; Update the error and move y if necessary
  lda err
  clc
  adc dy       ; err += dy
  sta err
  cmp dx        ; If err >= dx...
  bcc skip_y_step
  sec
  sbc dx       ; ...subtract dx from err
  sta err
  lda yy       ; ...and increment/decrement y
  clc
  adc sy
  sta yy
skip_y_step

  ; Increment or decrement x
  lda xL
  clc
```

```
    adc sx
    sta xL
    bcc no_carry
    inc xH          ; Handle carry if necessary
no_carry

    ; [check if we have reached the end point]
```

If the line is more horizontal (dx > dy), use the x_loop routine:

- Draw the current point (by calling PLOT)
- Update err: add dy
- If err exceeds dx, then:
- Correct the error by subtracting dx
- Also take a vertical step (i.e., change yy)
- Take a step horizontally anyway (update xL)
- Check if we have reached the destination (i.e., if xL = x2 and yy = y2)
- Otherwise, repeat everything

Loop for dominant lines in y (y_loop)

```
y_loop
    jsr PLOT        ; Draw the point

    ; Update the error and move x if necessary
    lda err
    clc
    adc dx          ; err += dx
    sta err
    cmp dy          ; If err >= dy...
    bcc skip_x_step
    sec
    sbc dy          ; ...subtract dy from err
    sta err
    lda xL          ; ...and increment/decrement x
    clc
    adc sx
    sta xL
    bcc skip_x_step
    inc xH          ; Handle carry if necessary
skip_x_step

    ; [movement along y and end point control]
```

Works in much the same way as x_loop, but with the roles of X and Y reversed.

PLOT: Function to draw a single pixel

```
PLOT
    lda xL
    and #7          ; Keeps only the 3 least significant bits (0-7)
    tax
    lda #0
    sta loc
    lda xL
    and #$f8        ; Keeps only the 5 most significant bits (multiples of 8)
    sta store
    lda yy
    lsr             ; Shift right (divide by 2)
    lsr             ; Shift right again (divide by 2 again)
    lsr             ; Shift right again (divide by 8 in total)
    sta loc+1       ; Stores in loc+1 (upper part of the address)
    lsr             ; Shifts right again (divides by 16 in total)
    ror loc         ; Rotates right with carry in loc
    lsr             ; Shifts right again (divides by 32 in total)
    ror loc         ; Rotate right with carry into loc again
    adc loc+1       ; Add loc+1 to register A
    sta loc+1
    lda yy
    and #7          ; Keep only the 3 least significant bits (0-7)
    adc loc         ; Add to the value of loc
    adc store       ; Add the value of store (position x)
    sta loc         ; Stores the final result in loc
    lda loc+1
    adc xH          ; Adds the high byte of x (for coordinates x > 255)
    adc bmpage      ; Adds the offset of the bitmap page
    sta loc+1       ; Stores the final result in loc+1
    ldy #0
    lda (loc),y
    ora BITMASK,x   ; Performs an OR with the bit from the BITMASK table
    sta (loc),y
    rts
```

As we know, in the C64 the resolution of the bitmap page is 320x200 pixels. This means that, in total, the screen consists of 64,000 pixels.

Bitmap mode allows you to draw pixel by pixel, but the video memory is organized in a non-linear way. The actual bitmap occupies 8000 bytes, where each byte represents 8 horizontal pixels, because each bit of the byte corresponds to a pixel that is on (1) or off (0).

So to turn on a single pixel, you need to find the memory address corresponding to the position (x, y) and you need to know which bit of that byte to turn on.

This requires a deep understanding of the hardware and memory organization. What could be implemented today with a simple call to a graphics function requires complex arithmetic calculations and bit-level manipulation on the C64. The PLOT function takes the coordinates (x, y), stored in the variables xL, xH (for x coordinates greater than 255) and yy, and calculates the corresponding memory address and the specific bit to be activated.

Analysis of the PLOT function

>Preparation of the bit to be modified

```
lda xL
and #7        ; Keeps only the 3 least significant bits (0-7)
tax           ; Transfers the value to register X
```

Let's start with the X coordinate, which can be any number from 0 to 319. But every 8 horizontal pixels are all within the same byte. So, if for example X is 14, the first 8 pixels (from 8 to 15) are in the same byte. The AND #7 command is a trick to say, "What is the position within that byte?" (i.e., X mod 8).

So, these first three instructions extract the remainder of the division of the x coordinate by 8. This value (from 0 to 7) indicates which of the bits within the video memory byte needs to be modified and is placed in the X register, because we will use X to retrieve the corresponding bit from a mask table (BITMASK).

>Calculating the memory address

```
lda #0
sta loc       ; Initializes the low part of the memory address
```

This part resets the low part of the address (called loc), because we are going to construct a 16-bit address that will tell us exactly where the byte is located in the bitmap memory.

```
lda xL
and #$f8      ; Keeps only the 5 most significant bits
(multiples of 8)
sta store     ; Temporarily stores
```

Now let's take the "coarse" part of the X. With AND #$F8 we keep only the high bits of xL, i.e. multiples of 8. This tells us which byte the pixel is in (not which bit), i.e. the number of the horizontal block we are working on.

>Calculating the vertical offset

```
lda yy
lsr           ; Shift right (divide by 2)
lsr           ; Shift right again (divide by 2 again)
lsr           ; Shift right again (divide by 8 in total)
sta loc+1     ; Store in loc+1 (upper part of the address)
```

Now comes the more technical and perhaps more surprising part. The C64's bitmap memory is not organized row by row as we would expect, but in blocks of 8 rows that follow each other in steps in memory.

We divide Y by 8 (with three shifts to the right) and the result is stored in the upper part of the address. This tells us which block of 8 rows the Y coordinate belongs to. These blocks are scattered throughout the video memory, so to know where the row begins, we also have to decipher the position within that block.

By saving this value in loc+1 (the upper part of the address), it is as if we were multiplying it by 256. This is due to the representation of 16-bit addresses in little-endian format: loc is the low byte (LSB) and loc+1 is the high byte (MSB).

>Multiplication by 320 (screen width in bytes)

```
lsr           ; Shift right again (divide by 16 in total)
ror loc       ; Rotate right with carry in loc
lsr           ; Shift right again (divide by 32 in total)
ror loc       ; Rotate right with carry in loc again
adc loc+1     ; Sum MSB of 256*int(y/8) + LSB of 64*int(y/8)
sta loc+1     ; MSB of 320*int(y/8)
```

Now comes the most complex part. To calculate the correct offset, we must multiply Y/8 by 40 (screen width in bytes), which is equivalent to 320 pixels:

320 = 256 + 64, so if loc+1 already contains Y/8 * 256, we just need to add Y/8 * 64.

This code is performing a series of operations to calculate Y/8 * 64 efficiently:

-The first lsr further divides by 2 (Y/16)

- ror loc rotates the least significant bit of A into the most significant bit of loc and shifts all bits of loc to the right
- The second lsr divides by 2 again (Y/32)
- Another ror loc rotates the carry into loc
- adc loc+1 adds this value to the high byte of the address

So in loc we have the low byte of Y/8 * 64, while in loc+1 we now have:

$Y/8 * 256 + Y/8 * 64 = Y/8 * 320$

This is the complete vertical offset to find the correct row in the bitmap.

>Calculating the low part of the address

```
lda yy
and #7        ; Keeps only the 3 least significant bits (0-7)
adc loc       ; Adds to the value of loc
adc store     ; Adds the value of store (x position)
sta loc       ; Stores the final result in loc
```

Now let's go back to Y: AND #7 gives us the row within the block of 8. We add this to the lower part of the address to get the precise row. Then we also add the value previously saved from X (store) to complete the horizontal position.

>Calculation of the upper part of the address

```
lda loc+1
adc xH        ; Adds the upper byte of x (for coordinates x > 255)
adc bmpage    ; Adds the offset of the bitmap page
sta loc+1     ; Stores the final result in loc+1
```

Here we take into account whether the X coordinate is so large that it exceeds 255 (and therefore whether it goes on two bytes). We also add the value of bmpage, which tells us where the bitmap memory begins (in our case it will be $2000, which corresponds to a bmpage of $20, or $6000, which corresponds to a bmpage of $60).

>Activation of the specific pixel via Indirect Addressing Y

```
ldy #0
lda (loc),y   ; Reads the byte pointed to by (loc)
ora BITMASK,x ; Performs an OR with the bit from the BITMASK table
sta (loc),y   ; Writes the modified byte to memory
rts           ; Returns from the subroutine
```

Finally, we turn on the pixel! Here, indirect addressing with index Y is used. We have the precise address in the bitmap memory and the bit we want to turn on.

- lda (loc),y: loads the byte containing our pixel
- ora BITMASK,x: turns on only that specific bit, leaving the others unchanged
- sta (loc),y: writes the byte back to memory
- rts: returns to the calling routine

The address calculation algorithm implements this formula:

$address = base + (y \text{ AND } 7) + ((y / 8) * 320) + (x / 8)$

Where:

- base is the base address of the bitmap memory (in this case $2000 or $6000)
- (y AND 7) is the row within the character
- ((y / 8) * 320) is the offset of the character row
- (x / 8) is the character column

>The BITMASK table

This is a table of 8 values containing bit masks for selecting a single bit within a byte:

```
BITMASK
  .byte %10000000, %01000000, %00100000, %00010000, %00001000, %00000100, %00000010, %00000001
```

For example, if x mod 8 = 3, the fourth value (%0001 0000) will be used to activate the fourth bit from the left in the byte.

Compiling the Assembly file

Once you have written the source file in Assembly, named for example "octahedron.asm", you need to assemble it using the 64tass assembler.

Make sure you have 64tass installed (you can download it from: https://sourceforge.net/projects/tass64/).

Open a terminal or command prompt and type:

```
64tass -c -a octahedron.asm -o octahedron.prg
```

After loading the resulting octahedron.prg file onto the C64 (or emulator), simply type the command SYS 49152 to run the program.

Conclusion

We have seen how the use of bitmap mode allows direct control of pixels, which is essential for obtaining precise geometric figures. We have used techniques such as double buffering to avoid flickering and ensure smoother animation, and we have logically separated the calculation of coordinates from the actual drawing in order to optimize the available resources.

Of course, the entire work was conceived for educational purposes: many aspects can be optimized or extended, both in terms of performance and code organization. We therefore invite you to take it as a starting point for further exploration and improvement.

Although the rotation itself was delegated to an external program to simplify the logic and lighten the load on the CPU, the part written in Assembly handled rendering, frame management, and video synchronization in real time, showing how even a 6510 processor at just over 1 MHz can animate three-dimensional figures with surprising results.

And perhaps this is the timeless appeal of the C64: turning every limitation into a challenge, trying to get the most out of a machine born in an era when power mattered less than ingenuity.

Thank you for following me this far, and see you next time for another adventure with our C64!

Listing of the entire code in Assembly

You can find the complete listing at the following address:

- https://pastebin.com/E4E3wk15

Listing of the entire code in Python

You can find the complete listing at the following address:

- https://pastebin.com/kYjjnS5W

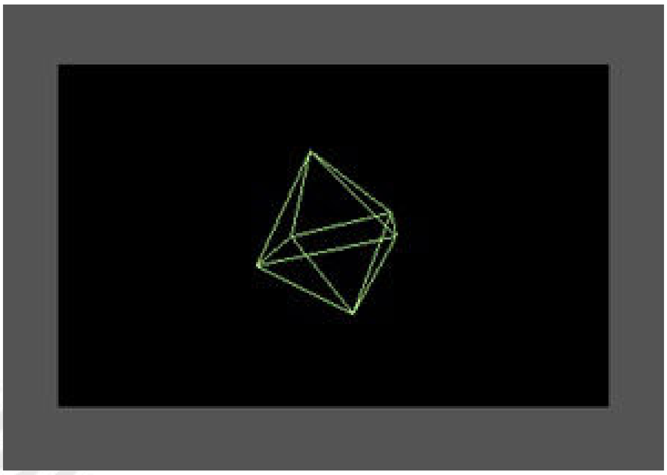


Fig.8 - Program running on C64

Bibliograhic References

Mathematics and Geometry

- E. Miglio, N. Parolini, A. Scotti, C. Vergara (2019). ***Matematica e Design***. Springer.
- P. Brandi, A. Salvadori (2004). ***Modelli matematici elementari***. Paravia-Bruno Mondadori.
- C. Petronio (2015). **Geometria e algebra lineare**. Società Editrice Esculapio.

Computer programming

- N. Ceder (2019). ***Python. Guida alla sintassi, alle funzionalità avanzate e all'analisi dei dati***. Apogeo.
- G. Naldi, L. Pareschi (2002). ***Matlab. Concetti e Progetti***. Apogeo.
- Raeto, Collin, West (1987). ***Programming the Commodore 64: The Definitive Guide*** (Rev. ed.). COMPUTE! Books Publication. Available here: https://archive.org/details/computes-programming-the-commodore-64-the-definitive-guide-revised-edition_202111
- S. Leemon (1987). ***Mapping the Commodore 64 & 64C***. COMPUTE! Books Publication. Available here: https://archive.org/details/Compute_s_Mapping_the_64_and_64C
- J. Butterfield (1986). ***Machine Language for the Commodore 64 and Other Commodore Computers*** (Rev. ed.). Prentice Hall Press. Available here: https://archive.org/details/Machine_Language_for_the_Commodore_Revised_and_Expanded_Edition/

Computer Graphics and Animations

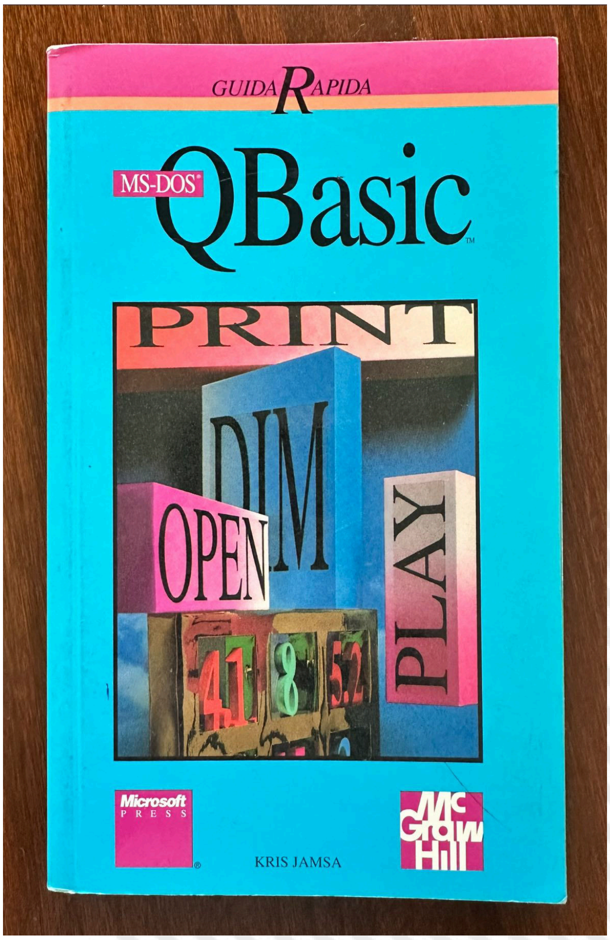
- J. Foley, A. van Dam, S. Feiner, J. Hughes (1995). ***Computer Graphics: Principles and Practice*** (2nd ed.). Addison-Wesley. Available here: https://archive.org/details/foley-james-d.-et-al.-computer-graphics-principles-and-practice-2nd-edition/
- S. J. Janke (2014). ***Mathematical Structures for Computer Graphics***. John Wiley & Sons. Available here: https://resource.laikipia.ac.ke/sites/default/files/Mathematical%20Structures%20for%20Computer%20Graphics.pdf

The RetroMagazine World Library

by Francesco Fiorentini

Title: Primo incontro con il computer
Subtitle: Il computer nella nostra vita di oggi e di domani
Author: Lorenzo Pinna
Publisher: Giunti-Nardini Editore
Year: 1984

Foreword by Piero Angela, as Lorenzo Pinna was one of Quark's collaborators.
A book rich in illustrations and fairly basic content to spread the idea of computers to an audience (the Italian one in the mid-1980s) that still saw computers as something futuristic but not feasible.
Space was also given to video games, with particular reference to Intellivision (Advanced Dungeons and Dragons, Burger Time, and Boxing), Atari (Crystal Castle, Ms. Pac-Man, Joust, Pole Position, and Mario Bros), and Colecovision (Looping and Mouse Trap).
(Italian)



Title: MS-DOS QBasic
Subtitle: Guida Rapida
Author: Kris Jamsa
Publisher: McGRAW-HILL Libri Italia srl
Year: 1992

A handy quick reference guide to all the instructions and functions of Microsoft QBasic. All entries include a brief description, complete syntax, parameters, and often a piece of code illustrating its use.
The guide also contains QBasic command line options and a general description of the language's data types, variables, and operators.
There is also a brief guide to running the QBasic environment and creating, editing, saving, and running programs.
The comparison table between QBasic and GW-Basic/Basica is very interesting.
(Italian)

The mysterious commands of MS-DOS: EDLIN

by Francesco Fiorentini

Probably almost all of us have had to deal with the MS-DOS operating system at least once in our lives. MS-DOS (acronym for Microsoft Disk Operating System) is an operating system developed by Microsoft starting in 1981 to run on PC-compatible computers. Unlike the more modern MS Windows, MS-DOS works through a character-based command line interface where users type commands directly.

One of the distinctive features of MS-DOS is undoubtedly its simplicity, with just a few commands that anyone can master fairly quickly. However, behind this simplicity lie a few small pitfalls. With this article, I would like to begin a short series of articles describing some of the more obscure DOS commands.

EDLIN, the MS-DOS text editor

The first command we will discuss in this brief overview is EDLIN (EDit LINes), a line editor.

Let's be honest, we've all tried running EDLIN at least once, but after a few unsuccessful attempts, we abandoned it in favor of the more convenient EDIT (or at least, I did). Now, at the ripe old age of 52, driven by a desire to leave nothing to chance, I want to give the much-maligned EDLIN another chance.

A bit of history...

EDLIN was created by Tim Paterson in just two weeks of development in 1980 for 86-DOS, based on the CP/M ED editor. Microsoft acquired 86-DOS and, after some further development, sold it as MS-DOS, so EDLIN was included in versions v1.0â€“v5.0 of MS-DOS. From MS-DOS 6 onwards, it was removed in favor of the full-screen MS-DOS Editor.

EDLIN is an external command* of MS-DOS:
EDLIN.COM - byte 8096.
Syntax: EDLIN filename
EDLIN cannot be launched without specifying a filename. If the file does not exist, it will be created; otherwise, it will be opened for editing.

*Remember that internal commands are those that reside in the COMMAND.COM file and are always available (COMMAND.COM is loaded into memory), while external commands are executable programs (with the .EXE or .COM extension) that are loaded only when requested.

When launched, EDLIN is rather laconic, which has probably contributed to its limited use and success. In fact, if you don't know EDLIN's internal commands, it is practically impossible to use it and do anything useful with it. So let's take a look at what these commands are and how they should be used.

EDLIN commands

The list of EDLIN internal commands for MS-DOS 2.0 and examples of syntax. Parameters in square brackets [] are optional.

- **(A)PPEND**: adds a specified number of lines to the file in memory - a [number]
- **(C)OPY**: copies a set of lines to a specific position - [line], [line], line c
- **(D)ELETE**: deletes a set of lines - [line], [line] d
- **EDIT**: corrects a line of text - [line]
- **(E)ND**: ends EDLIN and saves the file - e
- **(I)NSERT**: inserts lines into the text - [line] i
- **(L)IST**: displays the lines of text: [line], [line] l
- **(M)OVE**: moves a set of lines to a specific position - [line], [line], line m
- **(P)AGE**: [line], [line] p
- **(Q)UIT**: ends EDLIN without saving the file - q
- **(R)EPLACE**: replaces one string [string1] with another [string2] in a specified set of lines - [line], [line] r [string1] <CTRL+Z or F6> [string2]
- **(S)EARCH**: searches for a string in a specified set of lines: [line], [line] s [string]
- **(T)RANSFER**: transfers the contents of a file to the one we are editing - [line] t [file]
- **(W)RITE**: writes a number of lines currently in memory to disk - [number] w

Let's look at some of them in detail to understand how to use them.

(I)NSERT

The insert command inserts lines into the text. Each line of text can contain up to 253 characters, and you must press <ENTER> to finish it; if you don't, the line will not be saved. To end the INSERT command, press CTRL+C.

```
*i
1: test line 1
2: line 2
3: line 3
4: line 4
5: last line 5<CTRL+C>
6:
```

(S)EARCH

The search command searches for the specified string in the file.

```
*s4
4: line 4
```

(R)EPLACE

Replaces string1 with string2 in the specified range of lines.

```
*3,4rline<CTR+Z>line
3: line 3
4: line 4
```

(L)IST

Displays the contents of the file in memory.

```
*l
1: line 1 test
2: line 2
3:*line 3
4: line 4
5: line 5 last
```

Note that the asterisk indicates the current line.

The difference between LIST and PAGE is that with the latter, the last line displayed becomes the current one.

(M)OVE

Moves entire lines to a new position, before the specified line.

```
1,2,4m
*l
1: line 3
2:*line 1 test
3: line 2
4: line 4
5: line 5 last
```

We have moved lines 1 and 2 before line 4.

EDIT

The EDIT command is used to correct a line of text; to do this, simply indicate the line number:

```
*2
2:*line 2
2:*
```

At this point, all you have to do is type the new text and press <ENTER> to replace it in the file.

However, the EDIT command also has other features that can be activated using the function keys:

- F1: displays the text of the line to be edited one character at a time
- F2: copies the text of the line to be edited until it encounters the specified character. E.g. F2 and then e will be displayed: lin
- F3: displays the text of the line to be edited until the end
- F4: deletes the text of the line to be edited until it encounters the specified character. E.g. F4 and then e will display: ea 2

(T)RANSFER

Adds the contents of the specified file to the one being edited.

***5t testo1.txt** - adds all the contents of the file testo1.txt to the current one before line 5. The file testo1.txt, in this case, is located in the same directory.

(W)RITE and (A)PPEND

WRITE and APPEND deserve special mention. Once EDLIN is launched, its first task is to load the file into the computer's memory using up to 75% of the available memory. If the file is longer than 75% of the available memory, the WRITE command can be used to write some lines contained in memory to disk, thus freeing up a portion that can be used by APPEND to load new lines. APPEND and WRITE can only be used if the file we are editing is too large to be contained in memory.

Peculiarities of EDLIN

Despite its rudimentary interface, EDLIN had a peculiarity that was perhaps underestimated. It could be used within scripts. This feature, if exploited properly, would certainly have rescued EDLIN from oblivion.

Bibliography
- Steven Simrin (1986). ***Sistema Operativo MS-DOS***. Tecniche Nuove.

Online resource for testing MS-DOS 2.11:
- https://www.pcjs.org/software/pcx86/sys/dos/compaq/2.11/

Random permutations for a question (for C64)

by Eugenio Rapella

If our Commodore 64 is required to shuffle a deck of cards, mix up the answers to a multiple-choice question, or similar tasks, well... it needs to generate a random permutation of numbers from 1 to N.

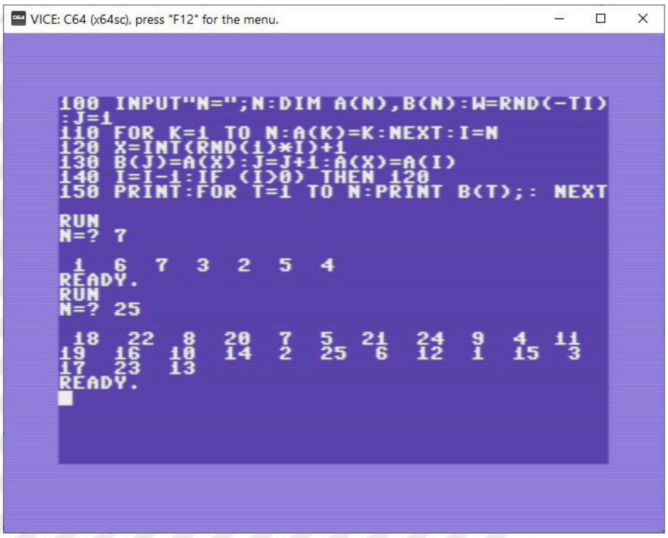
The C64 does this very well in six instructions:

```
100 input"n=";n:dim a(n),b(n):w=rnd(-ti):j=1
110 for k=1 to n:a(k)=k:next:i=n
120 x=int(rnd(1)*i)+1
130 b(j)=a(x):j=j+1:a(x)=a(i)
140 i=i-1:if (i>0) then 120
150 print:for t=1 to n:print b(t);: next
```

At 100, the number N of elements to be shuffled is requested and two vectors A and B are sized (for convenience, we will not use A(0) and B(0)). At 110, vector A is loaded with the numbers from 1 to N: A(1)=1, A(2)=2, â€¦, A(N)=N; index I is initialized with the value N.

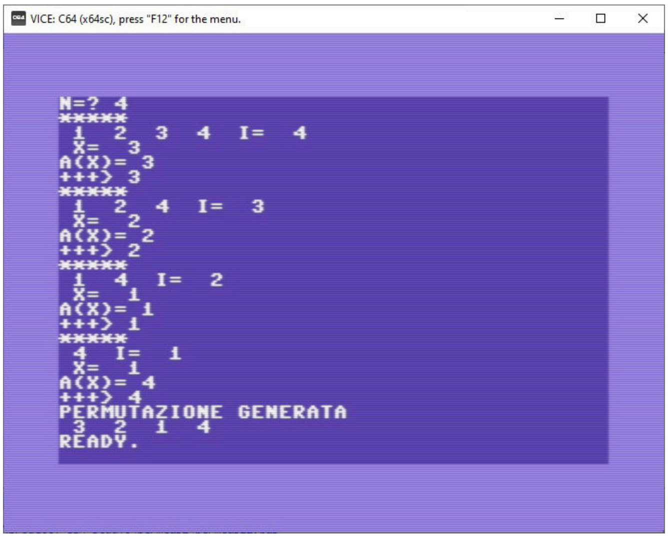
The actual program begins at 120: a random integer X between 1 and I is generated, therefore, at the beginning, between 1 and N. The first element of the random permutation will be A(X), which is copied to B(..). A(X) is replaced by A(I), I is decremented, and the program returns to 120. In this way, the first I positions in vector A are always occupied by elements that have not yet been drawn.

Here is the listing and a couple of program outputs:



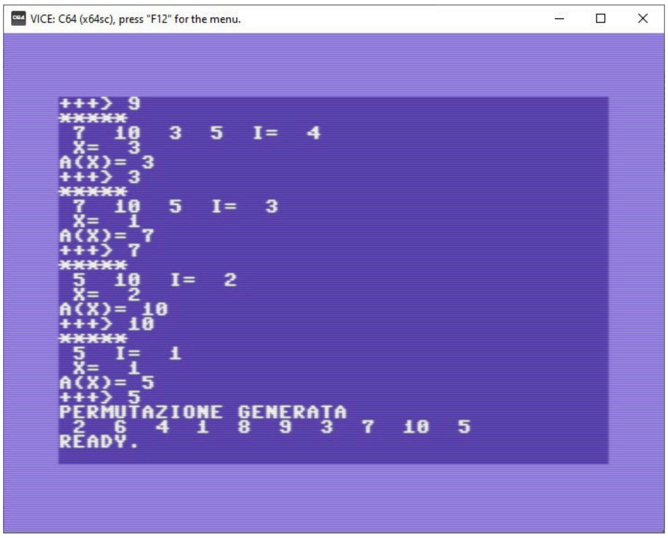
```
100 INPUT"N=";N:DIM A(N),B(N):W=RND(-TI)
:J=1
110 FOR K=1 TO N:A(K)=K:NEXT:I=N
120 X=INT(RND(1)*I)+1
130 B(J)=A(X):J=J+1:A(X)=A(I)
140 I=I-1:IF (I>0) THEN 120
150 PRINT:FOR T=1 TO N:PRINT B(T);: NEXT
RUN
N=? 7

 1  6  7  3  2  5  4
READY.
RUN
N=? 25

 18  22  8  20  7  5  21  24  9  4  11
19  16  10  14  2  25  6  12  1  15  3
17  23  13
READY.
```

The screen below shows the various steps; I have added some PRINT statements to the program: the first I elements of vector A are printed, along with the current value of I and the random value X between 1 and I.



```
N=? 4
*****
 1  2  3  4  I=  4
 X=  3
A(X)= 3
+++> 3
*****
 1  2  4  I=  3
 X=  2
A(X)= 2
+++> 2
*****
 1  4  I=  2
 X=  1
A(X)= 1
+++> 1
*****
 4  I=  1
 X=  1
A(X)= 4
+++> 4
PERMUTAZIONE GENERATA
 3  2  1  4
READY.
```

Below is the final part of the random generation of a permutation of ten elements:



```
+++> 9
*****
 7  10  3  5  I=  4
 X=  3
A(X)= 3
+++> 3
*****
 7  10  5  I=  3
 X=  1
A(X)= 7
+++> 7
*****
 5  10  I=  2
 X=  2
A(X)= 10
+++> 10
*****
 5  I=  1
 X=  1
A(X)= 5
+++> 5
PERMUTAZIONE GENERATA
 2  6  4  1  8  9  3  7  10  5
READY.
```

We can use the random permutations generated by the C64 to address a question proposed many years ago in the "Mathematical Games" column of the magazine "Le Scienze" (the Italian version of "Scientific American").

Here is the question:

Determine a nine-digit integer N consisting of a permutation of the digits 1 to 9 so that, for every integer K from 1 to 9, the number formed by the first K digits is a multiple of K. For example, the number 123456789 could be a good candidate for the solution: 1 is, of course, a multiple of 1; 12 is a multiple of 2; 123 is a multiple of 3, but, alas, 1234 is not a multiple of 4: 123456789 is therefore not a solution.

If you want to try to solve the problem without the help of the C64, you should know that there is only one solution (but we will come back to this), it is not immediate and requires at least a minimum knowledge of divisibility criteria (such as: an integer is a multiple of 3 if and only if the sum of its digits is a multiple of 3).

There are 9!=362880 permutations of the digits from 1 to 9, which is a bit too many for our Commodore. However, we can proceed with a significant reduction by noting that:

- The units digit of even numbers is even, which means that the digits 2, 4, 6, and 8 will go in the second, fourth, sixth, and eighth positions (not necessarily in that order).
- The units digit of multiples of 5 is either 0 or 5. Since "0" is not among the available digits, the digit in the fifth position must be 5.

Our number N will therefore be of the type DPDP5PDPD, where D stands for an odd digit other than 5 and P replaces an even digit. The permutations of the four available odd digits (1, 3, 7, 9) are 4!=24, and the same number are those of the even digits (2, 4, 6, 8). By "fitting" each of the former with each of the latter, we obtain 24*24=576 candidates from which to draw the solution.

Our C64 manages with a dozen instructions:

```
100 dim n(9):n(5)=5:x=rnd(-ti)
110 d$="1379":p$="2468":m=0
120 if m=1 then 150
130  m=len(d$):h=int(rnd(1)*m)+1:n(-(m=1)-(m=2)*3-(m=3)*7-(m=4)*9)=val(mid$(d$,h,1))
140 d$=left$(d$,h-1)+right$(d$,m-h):goto 120
150 m=0
160 if m=1 then 190
170 m=len(p$):h=int(rnd(1)*m)+1:n(-(m=1)*2-(m=2)*4-(m=3)*6-(m=4)*8)=val(mid$(p$,h,1))
180 p$=left$(p$,h-1)+right$(p$,m-h):goto 160
190 nt=nt+1:v=100*n(1)+10*n(2)+n(3):for k=3 to 8:f=v/k:print">";nt,v
200 if int(f)<>f then 110
210  v=v*10+n(k+1):next  k:print"ok  >  ";v:print"found with ";nt,"attempts"
```

At 100, the vector N(..) is sized, where the digits of the potential solution will be inserted as they are found. For convenience, we will not use N(0); as mentioned above, the digit 5 will go in the fifth position: N(5)=5. The instruction x=rnd(-ti) is intended to initialize the random number generator so that the same sequence is not always generated at each RUN.

At 110, D$ will be the string of odd digits and P$ the string of even digits.

Let's go to 130: at the beginning, D$="1379", so M=len(D$) assigns the value 4 to M, the current length of the string D$. H is a random integer between 1 and M, so at the beginning, H is between 1 and 4.

The index of the vector N(..) looks a bit complicated: -(M=1)-(M=2)*3-(M=3)*7-(M=4)*9.

In C64 Basic, an expression of the type "A=B" has a value of -1 if true, 0 if false. This means that the above expression will take one of the values 1,3,7,9 depending on the current value of M. Since at the beginning M=4, it will be N(9) that will take the value VAL(MID$(D$,H,1)), i.e., a random digit chosen in D$.

The effect of instruction 140 is to remove the chosen digit from D$ and then proceed to a new draw. Suppose, for example, that at the beginning, the random number is H=3; VAL(MID$(D$,H,1)) is 7: the third digit of the string D$="1379". Let's see how 140 modifies D$ in our example: LEFT$(D$,H-1) provides the first H-1 characters, i.e. the first two "13"; RIGHT$(D$,M-H) returns the last M-H, i.e. 4-3=1 characters of D$, so only the last character "9".

Again in 140, these two segments are placed side by side, so the new D$ becomes "139" and the effect is to remove the randomly selected digit from the string.

We then return to 120. Let's continue with our example; now D$ is "139" and M=3. H becomes a random number between 1 and 3; let's assume it is 3, then N(7) becomes 9 because "9" is the third character of the current D$. LEFT$(D$,H-1) is LEFT$(D$,2) or "13" while RIGHT$ (D$,M-H) is RIGHT$(D$;0) or a null string as it is formed by the last "zero" characters of D$. Conclusion: the new D$ becomes "13" and we return to 120.

When D$ is reduced to a single character (M=1 at instruction 120), this is necessarily randomly selected and assigned to N(1). Upon returning to 120, we move on to 150 and the vector N(..) will contain a random permutation of the digits "1379" positioned successively in N(9), N(7), N(3), and N(1).

The three instructions 160, 170, and 180 repeat the same procedure for even digits (as can be seen, P$ replaces D$); when we reach 190, the vector N(..) contains a random permutation of the digits from 1 to 9 of the type DPDP5PDPD.

At 190, the variable NT, initially zero, is incremented and contains the "number of attempts" made up to that point. At this point, the number consisting of the generated permutation is reconstructed in the variable V. Note that, given how the vector N(..) is constructed, the number formed by the first two digits is definitely a multiple of 2 (in N(2) there is always an even digit) and the number consisting of all the digits is, in any case, a multiple of 9 (a number is divisible by 9 if, and only if, the sum of its digits is divisible by 9. The sum of the digits of any permutation of 1,2, .., 9 is 45, which is a multiple of 9).

Since the number formed by the first digit alone is obviously a multiple of 1, verification of the property required in the problem can be limited to values of K from 3 to 8 (in fact, the number formed by the first five digits is also automatically a multiple of five, but checking when K is five and then skipping the verification would be more costly in terms of both code length and processing time).

Therefore, at 190, V initially takes the value of the number formed by the first three digits. Then, for K from 3 to 8, to see if the number formed by the first K digits is a multiple of K, we check if the integer part of V/K coincides with V/K. For example, if V=147 and K=3, the value of F is 147/3=49, which coincides with its integer part; if, on the other hand, V=143, we have F=143/3=47.666... and INT(F)=47 is different from F.

As soon as INT(F)<>F, it means that V is not the solution, so we return to 110 and start all over again.

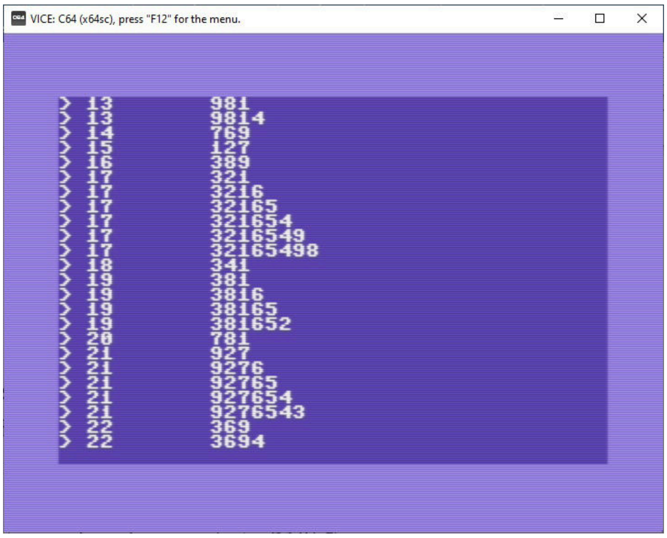
At 210, a new digit is added to V (the 4th, 5th, .., 8th) and the verification continues. The PRINT at 190 shows us the attempt number and the "partial" value of V.

For example, if at attempt number 19, the permutation generated is N(..)=381652947, the screen will display:

```
-      19     381
-      19     3816
-      19     38165
-      19     381652
-      20     ...
```

because 381 passes the first test: it is a multiple of 3; 3816 is a multiple of 4, 38165 is a multiple of 5, but 381652 is not a multiple of 6, so the C64 will move on to a new attempt.

When, for all K values from 3 to 8, things go well, we do not go to 110 and our C64 prints the solution (inst. 210).



```
> 13          981
> 13          9814
> 14          769
> 15          127
> 16          389
> 17          321
> 17          3216
> 17          32165
> 17          321654
> 17          3216549
> 17          32165498
> 18          341
> 19          381
> 19          3816
> 19          38165
> 19          381652
> 20          781
> 21          927
> 21          9276
> 21          92765
> 21          927654
> 21          9276543
> 22          369
> 22          3694
```

The probability of getting the right permutation is very low: P=1/576; the number of trials to be performed to obtain the first success is a "random variable" (a number that ... depends on chance) whose mean value is 1/P, the standard deviation is the square root of (1 - P)/(P*P). In our case, this means that we have to wait, on average, for 576 attempts, but with a standard deviation of about 575.5, the number of trials is very, very variable. As I write this article, I am repeatedly running the program using the VICE emulator. On the first RUN, the solution was found in 105 attempts, on the second RUN in 257, but today I am very lucky. Yesterday, for example, I closed the emulator when, after more than 1500 attempts, the C64 was still searching. Moral: press RUN and go for a walk, then... we'll see.

I submitted the question to ChatGPT, which instantly produced the solution. When I told my Commodore about this, well, he was very upset; he cheered up a bit when I explained that artificial intelligence hadn't calculated the solution, it had simply retrieved it from its huge database (this question is quite well known).

I tried asking ChatGPT if it was able to prove the uniqueness of the solution. The artificial intelligence replied that it was complicated and tried to propose a code (with some nonsense) to review all the possibilities in order to verify whether the solution was unique.

As you can imagine, the C64 took up the challenge: the following program is a variation of the previous one.

```
100 dim n(9),pe$(24),s(5),t(5):n(5)=5:d$="1379":p$="2468"
110 for k=1 to 24:read pe$(k):next
120 data"1234","1243","1324","1342","1423","1432"
130 data"2134","2143","2314","2341","2413","2431"
140 data"3124","3142","3214","3241","3412","3421"
150 data"4123","4132","4213","4231","4312","4321"
160 for x=1 to 24:for y=1 to 24
170 for k=1 to 4:m=val(mid$(pe$(x),k,1))
180 n(-(m=1)-(m=2)*3-(m=3)*7-(m=4)*9)=val(mid$(d$,k,1))
190 next k
200 for k=1 to 4:m=val(mid$(pe$(y),k,1))
210 n(-(m=1)*2-(m=2)*4-(m=3)*6-(m=4)*8)=val(mid$(p$,k,1))
220 next k
230 nt=nt+1:v=100*n(1)+10*n(2)+n(3):for k=3 to 8:f=v/k:print">";nt,v
240 if int(f)<>f then 260
250 v=v*10+n(k+1):next k:s(z)=v:t(z)=nt:z=z+1
260 next y:next x
270 print:print"number of solutions found ";z:print
280 for k= 0 to z-1:print"solution > ";s(k)
290 print"(at perm. n.";t(k);"of 576)":next
```

If desired, our Commodore is capable of generating all permutations of N objects (I discussed this in another article in this magazine), but here we are using the 24 permutations of four objects. The number is so small that it is more convenient and quicker to supply them directly to the C64: they are loaded into the PE$(..) vector using READ PE$(K) in instruction 110.
At the end of the FOR-NEXT cycle, we will have PE$(1)=”1234”, PE$(2)=”1243”, …, PE$(24)=”4321”.
At 160, two nested cycles open: the variable X uses the 24 permutations to generate the permutations of the odd digits; for each X, Y reuses them to generate the permutations of the even digits. In this way, the program reviews all 576 candidate solutions.
Since the goal is not only to find the solution, but also to verify its uniqueness, when a solution is found, it is loaded into the vector S(..) (instruction 250) and the program does not stop but continues the search.
Initially, the vectors S(..) and T(..) were sized to accommodate up to six different solutions, but at the end of processing, we will find that the variable Z, which counts the number of solutions, will have a value of 1. Therefore, only S(0) will have been used, and T(0) will contain the order number of the permutation that generated the unique solution.

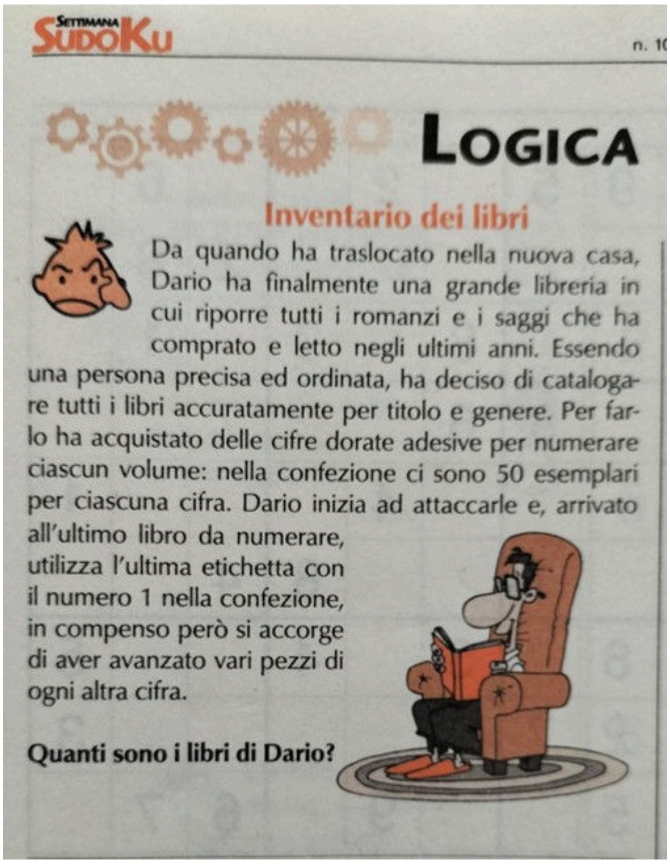
Ultimately, as far as uniqueness is concerned, ChatGPT threw in the towel, but not the C64!

I really think our Commodore deserves a standing ovation!

Dario's books, AI, and the C64 (for beginners)

by Eugenio Rapella

Each issue of the weekly Italian magazine "Settimana SudoKu" has a page dedicated to "Logic and Rebus." Here is one of the two problems proposed in issue no. 1041:



SETTIMANA SUDOKU

n. 10

LOGICA

Inventario dei libri

Da quando ha traslocato nella nuova casa, Dario ha finalmente una grande libreria in cui riporre tutti i romanzi e i saggi che ha comprato e letto negli ultimi anni. Essendo una persona precisa ed ordinata, ha deciso di catalogare tutti i libri accuratamente per titolo e genere. Per farlo ha acquistato delle cifre dorate adesive per numerare ciascun volume: nella confezione ci sono 50 esemplari per ciascuna cifra. Dario inizia ad attaccarle e, arrivato all'ultimo libro da numerare, utilizza l'ultima etichetta con il numero 1 nella confezione, in compenso però si accorge di aver avanzato vari pezzi di ogni altra cifra.

Quanti sono i libri di Dario?

Since moving into his new home, Dario finally has a large bookcase in which to store all the novels and essays he has bought and read in recent years. Being a precise and orderly person, he decided to catalog all the books carefully by title and genre. To do this, he bought some gold adhesive numbers to label each volume: there are 50 of each number in the pack. Dario starts sticking them on and, when he gets to the last book to number, he uses the last label with the number 1 in the pack, but then he realizes he has several pieces of each other number left over.

How many books does Dario have?

If you want to try to solve the question, stop reading and arm yourself with pen and paper (in fig. 2, the solution as it appears in the magazine... upside down, as tradition dictates!).

Our C64 only needs a handful of instructions:

```
100 a=1:print chr$(5)
110 n=a
120 k=int(n/10):c=n-k*10:if c=1 then t=t+1
130 if t=50 then print"n. libri:";a:end
140 n=k:if n>0 then 120
150 a=a+1:goto 110
```

Let's start with A=1. At 110, the current value of A is transferred to the variable N; at 120, the rightmost digit of the current value of N is determined. If, for example, N=37, we will have N/10=3.7; K=INT(N/10)=3 and C=N-K*10=37-3*10=37-30=7.

Also at 120, if C=1, the total T of the "digits 1" counted so far is increased by one unit.

When T reaches the value 50, we have used the last label with the "digit 1" and A contains the order number of the last book: mission accomplished.

If T is still below 50, the truncated number of the last digit (K) takes the place of N and, if N has not yet been canceled, we return to 120 to isolate the next digit.

For example, if at a certain point, A=101, we will have:

110: N=101

120: K=INT(101/10)=10 and C=1. T is incremented

140: N becomes 10 > 0, so we return to 120

120: K=INT(10/10)=1 and C=0. T is not incremented

140: N becomes 1 > 0, so we return to 120

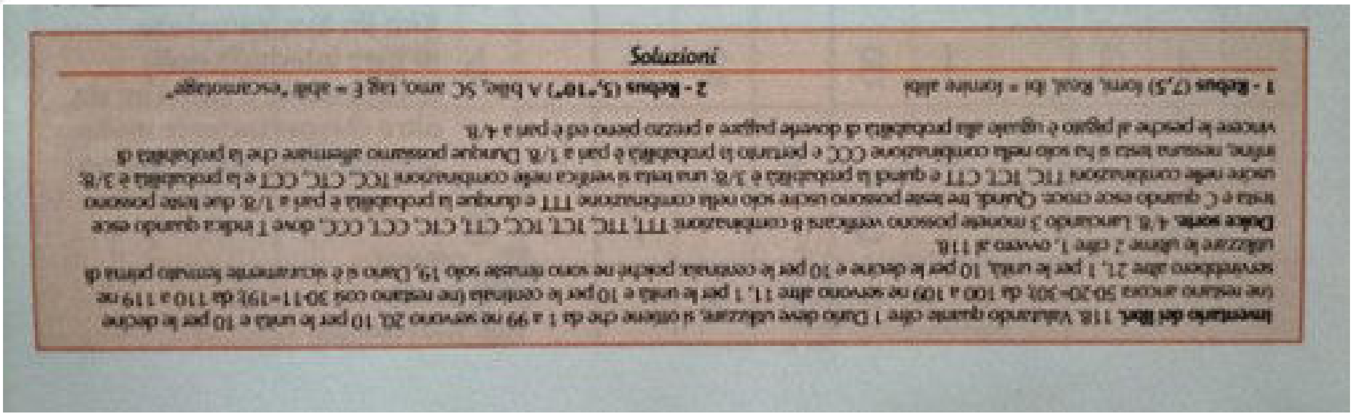
120 K=INT(1/10)=0 and C=1. T is incremented

140 N becomes 0: we do not go to 120 but continue

150 A becomes 102 and we return to 110 where N will become 102

As can be seen, T is incremented twice for the two digits 1 present in the number 101.

The problem states that Dario will have several labels left over with digits other than 1. With a simple modification to the previous program, the C64 can tell us how many labels were used for each digit when the fiftieth "digit 1"



Soluzioni

Fig. 2 - The solution to our question. Upside down, of course, as tradition dictates.

is reached. In the following program, the vector CF, sized at instruction 100, will contain the number of labels used for each digit: CF(0) for the digit 0, CF(1) for the digit 1, and so on up to CF(9) for the digit 9.

```
100 a=1:print chr$(5):dim cf(9)
110 n=a
120 k=int(n/10):c=n-k*10:cf(c)=cf(c)+1
130 if cf(1)=50 then 160
140 n=k:if n>0 then 120
150 a=a+1:goto 110
160 print"num. libri";a:print
170 for w=0 to 9:print w;cf(w):next
```

At 120, once the rightmost digit has been identified, the corresponding element of the vector is incremented by one. When CF(1) reaches the value 50, the program ends and at 170 the number of labels used for the various digits is printed (they will all be values below 50, as indicated in the text).

Of course, you can also try to solve this point without the help of our C64... it's up to you.

I then tried experimenting with some of the available "artificial intelligence" tools.
First, I simply copied and pasted the text of the problem. I won't go into detail about the answers I got; if you're interested, you can simply do the same thing and see the results (also because the outputs are always very long and detailed).

ChatGPT responds with:
--> Answer: 119 books
but the correct answer is 118 books.

DeepSeek accepts my input in Italian, but responds in English. First, it responds with 120, then, without me writing anything, it decides to check and writes "wait, this contradicts our step-by-step count...", "hmm, seems like an inconsistency...", "let me recount..." and finds the correct answer.

Gemini immediately finds the correct answer: **118**.

The request to report the number of labels used for each digit also provided interesting answers... you can try it and... have some fun.

Since we know that the solution is 118, I also tried asking the Artificial Intelligence a simpler question: "If you list the numbers from 1 to 118, how many times does each of the digits from 0 to 9 appear?" The interaction with DeepSeek was amusing, as it correctly reported all the digits except for the occurrence of the digit zero: it wrote 32 instead of 21. When I pointed out that 32 was a little too high, DeepSeek corrected the error and explained in detail why it had made the mistake.

I also asked it to solve the problem by producing code in C64 BASIC. After the first few answers, I had to point out that WHILE, WEND, and ENDIF are not part of our Commodore's vocabulary...

You're absolutely right! **Commodore 64 BASIC (version 2.0)** does not support the `WHILE...WEND` construct (that was introduced in later BASIC versions). Instead, we can use a `GOTO` loop with a conditional check. Here's

the corrected version.

Both DeepSeek and Gemini produced working code on the C64:

DeepSeek (also provides the correct values for the individual digits):

```
10 DIM D(9): REM DIGIT COUNTER (0-9)
20 N=1: REM CURRENT BOOK NUMBER
30 IF D(1) >= 50 THEN GOTO 110: REM STOP AT 50 '1'
40 S$ = STR$(N): REM CONVERT TO STRING (WITH SPACE)
50 FOR I = 1 TO LEN(S$)
60 C$ = MID$(S$, I, 1): REM EXTRACT CHARACTER
70 IF C$ <> " " THEN C = VAL(C$): D(C) = D(C) + 1
80 NEXT I
90 N = N + 1
100 GOTO 30: REM REPEAT
110 PRINT "total books:"; N-1
120 PRINT "digit count:"
130 FOR I = 0 TO 9
140 PRINT I; ":"; D(I)
150 NEXT I
```

Gemini:

```
10 REM COUNT THE DIGITS '1'
20 N = 0 : COUNT1 = 0
30 FOR I = 1 TO 200: rem check up to a reasonable limit
40   S$ = STR$(I) : rem Convert the number I to a string
50   FOR J = 1 TO LEN(S$)
60    IF MID$(S$,J,1) = "1" THEN COUNT1 = COUNT1 + 1
70   NEXT J
80   IF COUNT1 = 50 THEN N = I : GOTO 130: rem If COUNT1 is 50, store I and jump to the end
90 NEXT I
100 REM IF THE LOOP ENDS WITHOUT FINDING 50 '1'
110 REM IT MEANS THAT THE LIMIT OF 200 IS TOO LOW.
120 REM BUT IN THIS CASE IT SHOULD NOT HAPPEN.
130 PRINT "Dario's books are ";N
```

In these programs, the digits are extracted using the STR$() function (which transforms the number into a string of characters preceded by a "space") and the MID$() function, which extracts individual characters.

For more detailed comments on these codes, the easiest thing to do is to ask DeepSeek and Gemini again: they are not at all stingy with explanations (quite the contrary!).

Enjoy!

Generation 80

Diary of a digital native. Growing up and changing with technology.

by Carlo Nithaiah Del Mar Pirazzini

Reading Generazione 80 is like reading your own past. It's striking.

Biagio Alagna has created a book that flows pleasantly, made up of personal memories and profound reflections that we have all had. Those of us born in the years of the first home computers, noisy modems, and pixelated games.

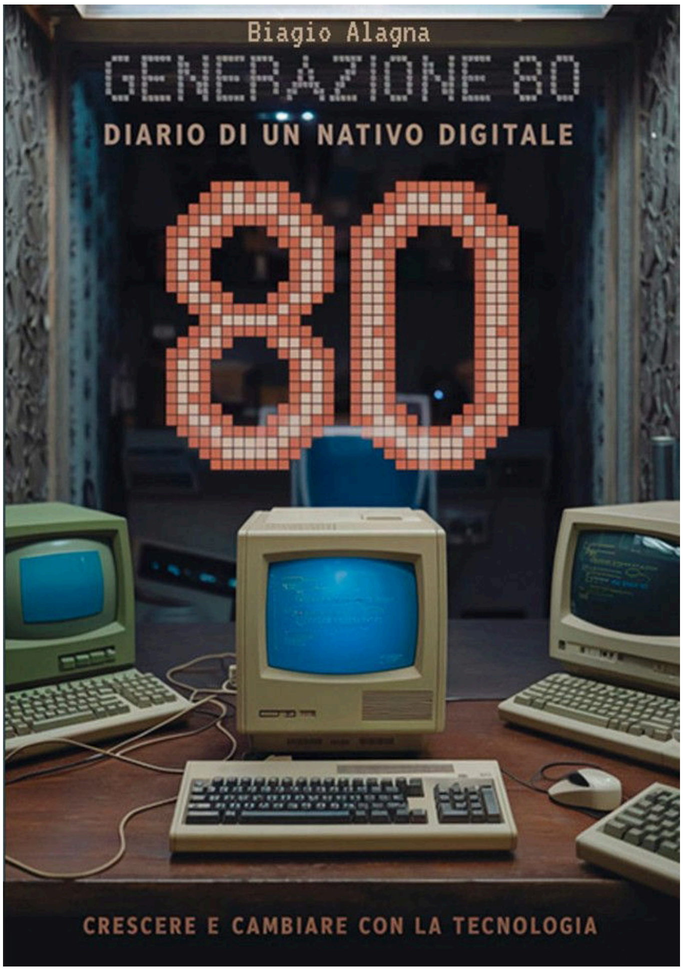
It is a valuable testimony for those who today take the "digital" world for granted and without enthusiasm. For these people, Generazione 80 is an exciting journey and almost a revolution.

Biagio describes his world at that time without dwelling on the past. He takes us on a journey through his experience of technology: it is not sterile, numerical, or descriptive, but full of life, energy, and concreteness.

The writing style is direct, always human, and engaging. I strongly believe that it is suitable not only for us nostalgics, but for anyone who wants to peek into the mind of a "digital dreamer" to understand where we come from.

He writes, 'The future will not be defined only by algorithms, but also by the stories of those who know how to use them with intelligence and heart'.

If you want to buy it, you can find it here:
https://unit.link/generazione-80


Biagio Alagna
GENERAZIONE 80
DIARIO DI UN NATIVO DIGITALE
80
CRESCERE E CAMBIARE CON LA TECNOLOGIA

Biagio Alagna - author of the book

The "new" Commodore

Interview with Christian “Peri Fractic” Simpson

by Carlo Nithaiah Del Mar Pirazzini

The birth of the "new" Commodore has sent shockwaves through the world of nostalgics, retro enthusiasts, and those who, like myself, lived through the glorious era of the C64 and Amiga machines.

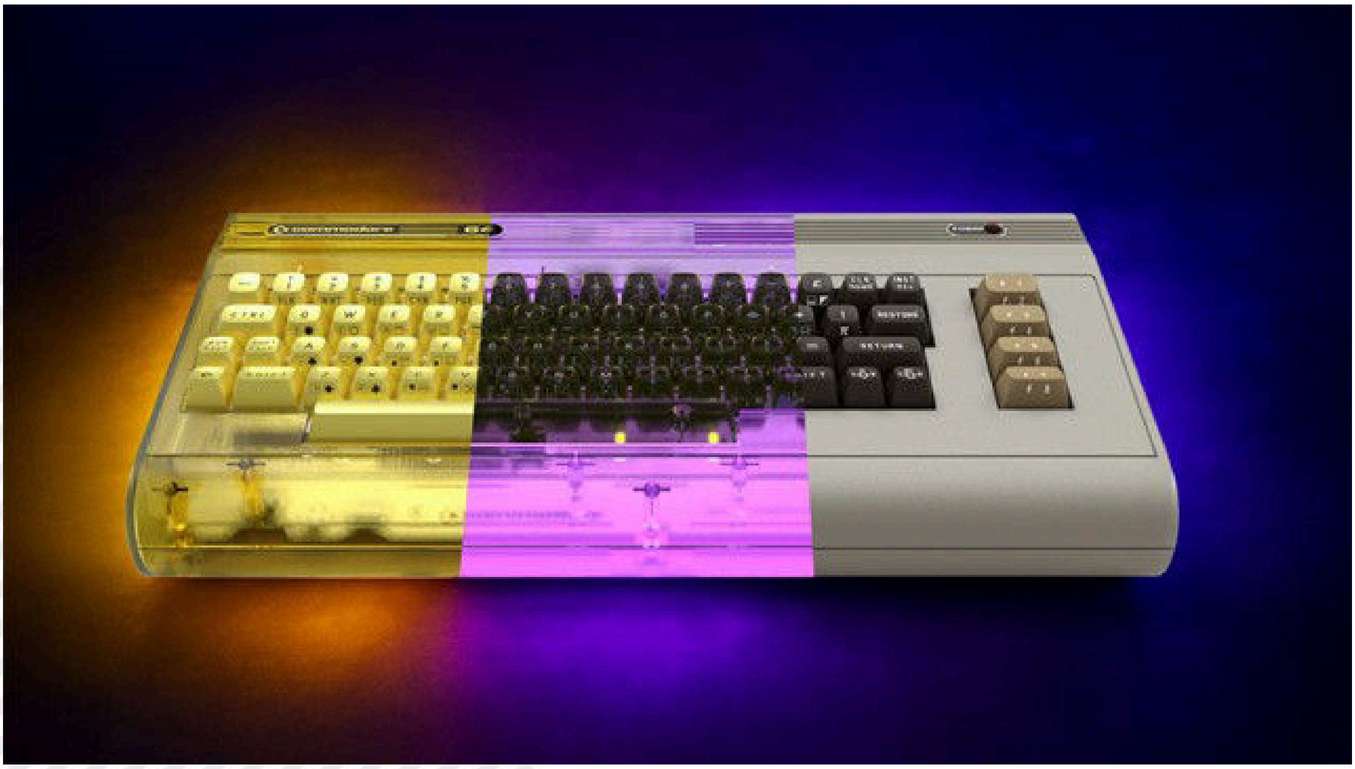
Christian "Peri Fractic" Simpson's new Commodore has already achieved incredible things. The brand has been taken over, the development team brings together the "old glories" of Commodore and a new circle of really talented retro developers, and the sales campaign for the new C64 has been a great success. Pre-orders for the C64 Ultimate (C64U) sold out in just a few hours.

We reached him by email and asked him a few questions about it, promising to get back in touch as soon as the Commodore 64 Ultimate was available for a "test drive." I'll leave you to the interview.

Christian “Peri Fractic” Simpson

How did this project come about?

Peri: It came from a childhood dream. Like many fans, I grew up with the C64, which sparked my love for technology. Years later, MyRetroComputer asked me to review the C64X, which had acquired the license for the project but



The new C64 Ultimate in three versions available on the official website

not the Commodore name. They were going to call it My64X because the trademark license was too expensive. But I know how to be very persuasive: I convinced them to stick with Commodore. My video got over a million views, and the rest is history. This momentum led to talks with the brand owners, and eventually, they invited me to acquire Commodore directly. This project is about doing things right, for all of us who grew up typing LOAD"*",8,1.

Tell us about the Commodore Team. What can you tell us beyond what we've seen on the website?

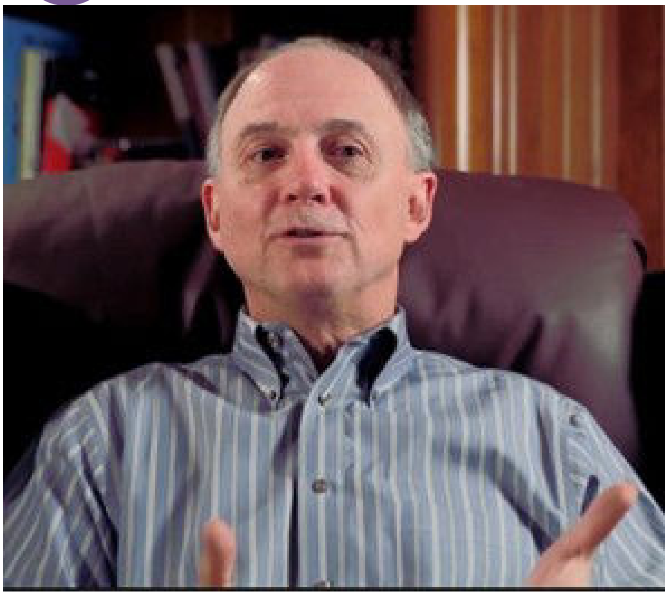
I wanted to create a team that truly understood the whole Commodore world: not just the brand, but its true soul. So we brought together legends like Albert Charpentier, the "father" of the C64, and Bil Herd, who built the C128. Added to these are new pioneers such as Jeri Ellsworth, Jim Drew, and others who have been shaping the retro scene for years. This is not just a technical team, but a family of Commodore enthusiasts, united by a mission to revive the brand with love and authenticity.

What else can you tell us about the Commodore 64 Ultimate? There's no emulation here, it's much more than that, right?

Leonard Tramiel

You've hit the nail on the head: it's not emulation, it's reproduction. Inside is Gideon's Ultimate 64 board, a faithful FPGA hardware reconstruction of the original C64 chipsâ€”SID, VIC-II, 6510â€”not just imitated, but electronically replicated. But we've gone further: it offers turbo modes up to 48 times the original speed, expandable RAM up to 16 MB, dual SID support (virtual or real), HDMI output, Wi-Fi game transfer, keyboard macros, and much more. All enclosed in a translucent, luminous case with

Leonard Tramiel will be part of the group alongside familiar faces such as Al Charpentier and Bil Herd, as well as David Pleasance, Alex Rozenblat, Jeri Ellsworth (above), and many others



Al Charpentier

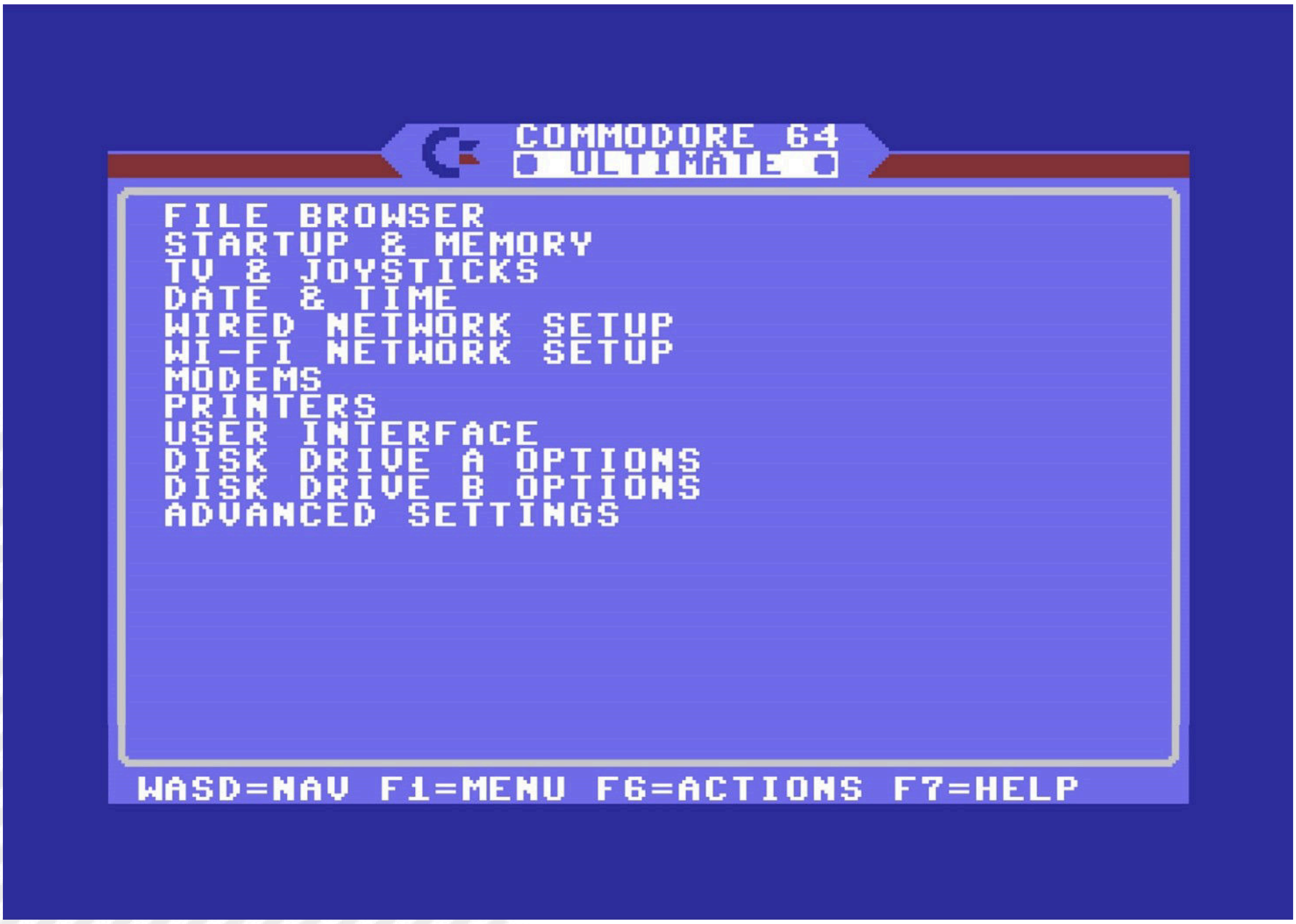
the world's first transparent PCB for a mechanical keyboard. You can even connect original cartridges, joysticks, CRTs, and floppy drives. It's a real Commodore, only... Ultimate.

What does the future hold for Commodore?

We call it the Digital Detox brand. In a world full of distractions, Commodore is once again synonymous with focused and joyful computing. We are creating retro-futuristic products that blend nostalgia and modern simplicity, such as the C64 Ultimate, a LEGO-style PC called Brixty Four, a distraction-free flip phone, and even AR gaming glasses. But more than products, Commodore's future is about community. We want fans to feel like they belong to this story, because in a way, they always have.

What do you think of the other Commodore companies involved? What will the relationship with Hyperion and Cloanto be like?

We respect anyone who has tried to keep the flame alive, even if their paths and visions differ. With Cloanto, we are conducting respectful negotiations to bring Commodore and Amiga under one roof. As for Hyperion, we will see how the legal situation develops. Our goal is not conflict. It is clarity. Fans deserve a clear and reliable source for official Commodore products. That's what we're building: together with the community, not against anyone.



The internal management menu of the Commodore 64 Ultimate

Interview with Luigi Bonifacino

20 questions to an important figure in retrocomputing: the electronics technician we turn to when you wish to resurrect our 8/16-bit systems

by David “Cercamon” La Monaca

Interviewing technical repair experts and retrocomputing enthusiasts is like digging into the beating heart of digital history. There is no need to explain the importance of the social media presence of these technicians, who come from different backgrounds and bring with them a wealth of extremely valuable knowledge. We often wonder if all this experience with the electronics used in our beloved home computers and vintage consoles can somehow be passed on through books, audio/video content, or in person, as was once the case in repair shops. We know that hardware built since the 1970s, although quite sturdy, is starting to show its age. When one of our 8/16-bit machines inevitably breaks down, we are immediately faced with the dilemma of whether or not it is worth having it repaired by a competent technician. This is also because it is not easy to find an expert on hand for a specific machine, and often the costs of spare parts (when available), labor, and shipping do not justify the repair. Yet these "angels" of retrocomputing often do not receive the recognition they deserve in the world of retrocomputing. So here, on RMW, is the first in a series of interviews dedicated to those who, in a sense, are behind the scenes of retrocomputing. A world of patience, intuition, tenacity, motherboards and circuits, memory cards and chips, burnt integrated circuits and blown capacitors, 'soured' batteries and tracks that are cut but invisible except under an electron microscope. Here are 20 questions designed to stimulate stories, anecdotes, and reflections from those who, with passion and sacrifice, manage to bring these wonders of the past back to life.

Q&A - 20 questions/20 answers

1. DLM - We would like to welcome and thank Luigi for kindly agreeing to this interview. First of all, we would like to ask you to briefly introduce yourself with a few notes about your personal history in relation to the world of electronics and computers/consoles.

Luigi Bonifacino in his laboratory

Hi, my name is Luigi Bonifacino, I live in Trapani and I was born in 1965, just in time to experience the home and personal computer revolution as a young boy. My history with electronics began when I was 13: I started attending a workshop run by a radio technician I knew, when TVs and radios still used vacuum tubes. After middle school, I attended the IPSIA institute, where I met my future employer: Commodore! From then on, it was all downhill. I worked for Commodore Italia (technical support) for about 10 years. In 1988, I moved to Milan to work in a large factory, and one day I met the owner of the famous "Circe Electronics," where I started working in my spare time. I was lucky enough to personally test the famous OC118 drive, a clone of the 1541, directly from the source. As a Commodore dealer and service provider, I learned the structure of many Commodore machines and peripherals, so I was able to start repairing them. These included ASSY motherboards, floppy disk drives, CRT monitors, various Amiga and PET systems, etc.

2. DLM - How did you become interested in repairing electronic boards and computers? Which machine are you most passionate about repairing and why?

I started when I was working in a Commodore store,

Luigi working at his desk

mostly for fun, but over time it became a job. The machine I like best is obviously the Commodore 64. It has a nice motherboard and it's now very easy for me to identify faults. In my experience, however, I've repaired everything: home computers, PCs, CRT monitors, external floppy disk drives, etc. And of course, I've dealt with many models from all brands. Commodore machines (64 and Amiga) are the most common, simply because of their popularity and the number of units sold.

3. DLM - What are the most common challenges you face in restoring and repairing retro hardware? What are the most frequent faults you encounter in your experience?

The worst challenges are restoring the corroded tracks on the Amiga, which are difficult to locate and repair. The most common faults are capacitors that leak acid and corrode nearby tracks. Another problem that often occurs is cold or poorly done soldering by others, chips burned out due to "long service" or inserted backwards by other repairers or owners and then left there without any sense.

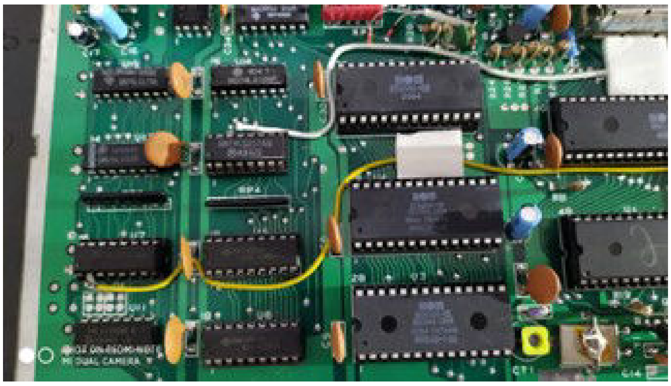
4. DLM - Do you have a "mission impossible" that you have successfully completed? Is there one that you are particularly proud of? Tell us about it briefly.

I have one that happened just recently: I had a dual Kickstart card for Amiga 500 in my hands where the second Kickstart was mounted on 4 EPROMs that turned out to be damaged. I could never find the right files, and I don't get along with software at all. So I asked around for experts to split the Kickstart into 4 parts, but everyone said something different and it still didn't work. Searching through various forums, I finally managed to find a suitable program (a DOS executable) that could split the file into as many pieces as I wanted, and only then was I able to bring this other piece of history back to life.

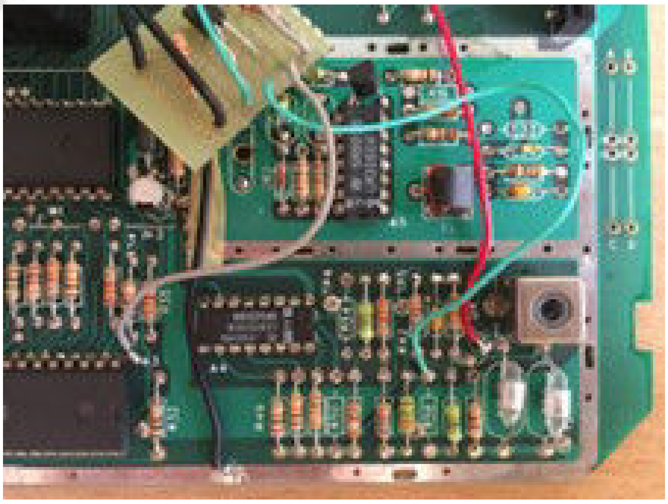
5. DLM - How do you get spare parts or original materials? How difficult is it to find them after more than 30/40 years?

I was a bit lucky. When the Commodore store where I worked closed, the owner gave me everything he had in stock in terms of components and spare parts. I still have some of it today. I buy RAM chips and other small components online from various websites, at least as long as I can find them.

6. DLM - What role does the retrocomputing community play in your business? Do you collaborate often? Have you created one around your hobby, perhaps with a YouTube channel or Instagram account?



Modification of RAM expansion to 64K on C16



Modification RF to composite video for Atari 2600

I don't have an official channel at the moment, I don't have the patience for these things. I manage to get some publicity on Facebook and in various groups dedicated to old computers and consoles. I know someone who is the administrator of a repair group that operates as a "storage facility" and when they have a lot of material, they send it to me.

7. DLM - In your experience, how important is it to preserve not only the devices, but also the software and documentation related to the hardware (technical diagrams, circuit drawings, tables, and data on the components used)?

It is very important. I have a vast archive of programs, electrical diagrams, and so on, as well as my personal notes. When I have a repair that requires some extra research, that is my first resource to draw on.

8. DLM - In recent years, have you noticed a resurgence of interest in retrocomputing/retrogaming among the younger generations and beyond? Who are the people who come to you for technical advice or repairs?

CF-IDE modification and video color repair on A1200

In my experience, only a few young people are interested in this field. Most are over 30/40 years old and even older, who are the former kids who used these machines the most when they first appeared on the market.

9. DLM - What is your opinion on emulation (software) or reproduction (FPGA) of old systems: does it help or harm the preservation of truly vintage systems?

In my opinion, abandoning the original systems is blasphemy: real machines must be preserved, period. I am very convinced of my position. I see hardware emulators such as The64, Amiga Mini, etc. around, and they are quite widespread, but they will never be comparable to the effect of using a real machine. On the other hand, I am in favor of software emulators, mainly because they are faster and more practical for testing new games or programs. I think reproduction via FPGA systems is fine, but I find the prices offered on various websites a bit exaggerated. For the same price range, I can find a valuable original machine, even if it's broken, and try to fix it and make it like new again.

10. DLM - If you could get your hands on and restore a single legendary piece for a retrocomputing museum, which one would you choose?

I would choose a C64 'biscottone', specifically the Assy 250425, which I consider cleaner than the previous model, where the 'video' part was crammed with integrated circuits. This version, on the other hand, has only one other chip (the 7501) in addition to the 6569 video chip. It's my favorite, we grew up together.

11. DLM - What basic hardware and software tools do you usually use to diagnose faults on vintage systems?

I was born in the analog era, so I use a simple analog and digital tester to check voltages, and sometimes I use an electron microscope to check tiny solder joints and broken tracks.

12. DLM - When working on a historic or valuable motherboard, do you prefer to keep it as is or make modifications/adaptations?

It all depends on what the customer asks me to do. In general, I try to preserve the original machine. In the past, there have been customers who, for example, wanted to revolutionize their Amiga 2000 with IDE cards, CF cards, accelerators, and more, changing the original machine a bit, but after closing it, you can't notice anything, so that's fine! Sometimes I have replaced an entire irreparable power supply with an ATX taken from a PC.

13. DLM - How do you deal with oxidation and degradation of electrical contacts after decades of inactivity? What maintenance techniques do you usually recommend to those who receive a machine after your repair?

Unfortunately, oxidation is a nasty "disease." You have to wash the board thoroughly and remove all the oxidation with special acids before proceeding with the repair. There are no particular recommendations for the user. It is assumed that after repair, a machine is in good/excellent condition and normal use by the user does not affect this problem.

14. DLM - Have you ever had to recreate or replicate components that are no longer in production? If so, how did you do it? What resources did you use to document yourself?

Fortunately, I have never had to recreate a component, but if it did happen, the only solution would be to cannibalize another similar machine. :-)

15. DLM - Do you use hardware emulators such as FPGAs or reverse engineering systems to test obsolete parts? What do you think are the fastest tools for diagnostics?

No, I've never used hardware emulators or other complex diagnostic equipment. In the past, to start diagnostics (e.g., on a faulty C64), I would obviously turn on the machine and if nothing appeared (black screen), I would start checking the various input voltages to verify that they were correct. Fortunately, today it's a little easier: I use the Diagnostic 64 cartridge, and if nothing appears even with it inserted, I use another little secret (which from now on will no longer be a secret): I insert the CLOWNS cartridge (the game). This cartridge is special because it works even if the ROM, RAM, and other parts of the computer are faulty. In practice, the cartridge bypasses almost all of the C64's hardware, so depending on the image that appears, I can tell from experience whether it is a RAM or ROM chip failure, etc.

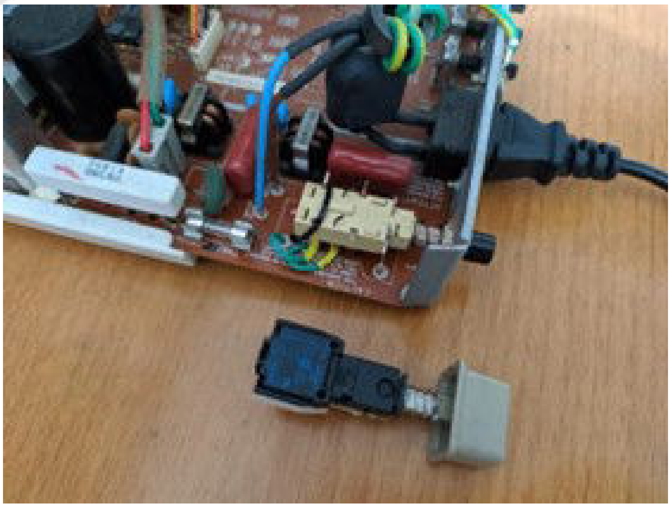
Of course, the diagnostics change depending on the type of computer. With an Amiga, it is obviously more complex, as it is a more advanced computer. In the case of a black screen, we also start by checking the input voltages and various parts of the motherboard. Sometimes, however, something appears on the screen with distorted colors, and then you can already tell that the problem lies in the video part (e.g., if a green screen appears, it is likely that the fault is in the RAM, and in this case, I use the DIAGNOSTIC ROM cartridge or "piggyback" on the RAM until I find the one that is not working and replace it). By repeating the process several times after repairing a problem, it is possible to completely resolve the fault and have a machine that is 100% functional again.

16. DLM - How do you manage the storage of data from fragile magnetic media (floppy disks, cassettes, hard disks)? Do you look for the most appropriate solution on a case-by-case basis, depending on the machine and system you have in front of you?

I have only had to store personal data, I have never had similar requests from customers. In order to preserve the data, I use normal CF Cards or TF Cards, where I have downloaded all my 5.25" floppy disks in binary/image format.

17. DLM - What is your opinion on the use of modern capacitors to replace vintage ones in repairs? In general, is it easier to carry out common repairs today than it

Repairing an Amiga 3000 with boot problems



Repairing the on/off button on the monitor

was when the systems were first sold?

In my opinion, they are the same, they have the same function, but they differ only in durability and shape. For example, horizontal capacitors are no longer available today, so to replace one, you have to lay the vertical capacitor down and extend one terminal to reach the opposite reoforo (capacitor terminal, Ed.).

18. DLM - What programming languages or software environments do you use to interface with legacy hardware, especially for firmware updates or ROM/EPROM-related updates or integrated circuit programming?

I don't use any programming languages. I hardly ever write code. I rarely use DOS to run specific programs to prepare binary files, while to write to ROM/EPROM I use Windows with the appropriate writing program.

19. DLM - How do you document your work: do you create diagrams, write guides, or share code on public repositories or multimedia content? Are you a technician who is "jealous" of your own techniques, or are you willing to share your knowledge and experience with others?

As I said before, I'm not very fond of "online" things. I prefer to keep my guides in my personal archive, but if any of my documents or technical diagrams are useful to someone, I would be willing to share them.

20. DLM - How much do you really enjoy performing repairs, diagnosing faults, and how much satisfaction do you get when you manage to solve a problem that initially seemed insurmountable? Have you ever had to give up because of too many defects in a system or because the cost was not worth the effort?

That's the most enjoyable part of the job: I have a blast seeing things repaired and working again. It's very satisfying. And when I have nothing to fix, I look for the impossible just so I can hold a soldering iron in my hand! I have rarely given up on malfunctioning machines. When it did happen, I gave up because the system in question had been too "tampered with" or because it really wasn't worth the effort!

Conclusions

We would like to thank Luigi once again for taking the time to talk to us and for opening the doors (often ajar) of the electronic technicians who, with immense passion and dedication, take on the task of reviving the vintage systems that we love and preserve so much. If you have machines and peripherals that need repairing or checking, you can write to retromagazine.redazione@gmail.com, referring to this interview.

We will be happy to forward your requests to Luigi. In future issues, we will meet other Italian and European electronics technicians who often deal with the repair and modification of computers and consoles belonging to the golden age of 8/16-bit. Their role is very important, so be sure to support the technicians near you!

LINKS AND BIBLIOGRAPHICAL REFERENCES

Amiga PCB Explorer
- https://www.amigapcb.org

Amiga Technical Resource
- http://amiga.serveftp.net

Pictorial C64 Fault Guide
- https://www.pictorial64.com

C64 circuit diagrams
- http://www.cbmitapages.it/tecnico/schemi.htm

C64 Repair Guide
- https://retro64.altervista.org/blog/guida-alla-riparazione-del-commodore-64/

C64 repair manual
- https://archive.org/details/manualediriparazionedelcommodore64

Japan episode 27: Gachagachagachagacha...pon!

by Michele Ugolini

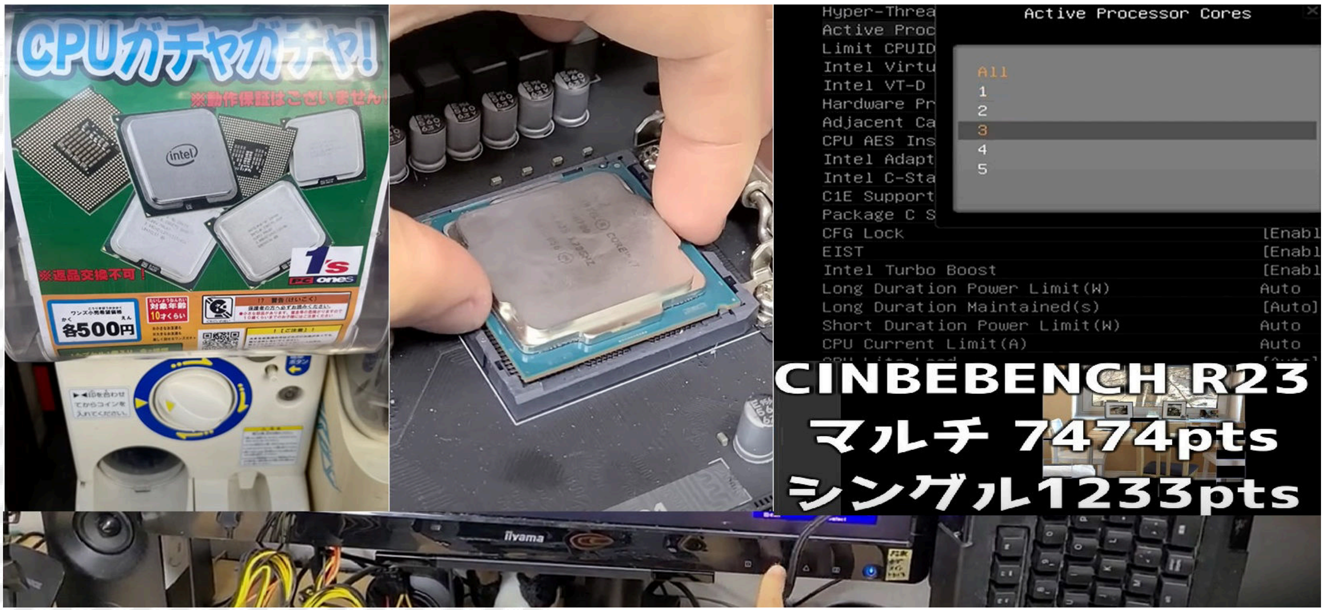
Small is better! Do you think this is a provocation? Absolutely not, it is an essential requirement for a population with a density of four million people around four square kilometers, near the Emperor's castle.
Small, tall cars, small, functional houses, high-precision miniaturized objects, desks with dividers to optimize concentration, oxygen tanks so you don't run out of air in the office before your coffee break... No, we haven't reached that point yet, but the happier, more spacious, and certainly more human-paced standards of living of the beautiful Italian country are very far from Japanese society.

However, regardless of this claustrophobic introduction, I too am surprised to see the vast quantity of collectible miniaturized objects in our beloved gachapon.
Considering that square footage is sacred in a Japanese home, collecting tiny objects seems to be an unassailable requirement.

In my previous articles, I have already listed various bizarre items, some exclusive, expensive, and some even dedicated to the adult world.
What could surprise us in this umpteenth article that will once again talk about these magical balls?
Well, I just discovered that in some vending machines, we can even buy CPUs!
The cost? Three or more 100-yen coins.
Here are two entertaining videos of this lucky buyer who found a CPU for only 500 yen.
The video shows the installation steps on the motherboard and the operating system.
There are also a few mild swear words and the discovery that this CPU is original, INTEL, but unfortunately damaged.
Clearly, an I7 processor for about five euros is unlikely to be perfect.
The user attempts to disable a CPU core, gets a good benchmark, and promises to return more often to Nihonbashi where he found it.

https://youtu.be/Tp9JAWBFsCA
https://youtu.be/zpU1WiyI3no

Let's continue our exploration.
If I told you that a ninja could be hiding behind a common USB port, would you believe me?
All the people who didn't believe me were wrong. A ninja can easily hide behind a USB port, or even behind a USB-C plug.
Want to exaggerate?
Okay, then, a really good ninja can even hide behind the plastic that wraps around pencil erasers.
Yes, it's true. Imagine a cell phone charging panel with USB ports, imagine its size of a few centimeters.
Now imagine that this panel is actually made of a fabric, like a curtain, with the design of that panel printed on it.
Imagine that behind this strange panel/curtain there is a ninja holding it up with his arms, hiding behind this bizarre curtain, panel, USB port, or, if you like,



visual hallucination, typical of ninja techniques.
I am attaching some of these funny camouflage proposals from various ninjas. Could it be interesting? Clearly, it is a strange object.
I definitely want it.
The manufacturer says: "Know that finally, anyone who has longed to live the life of a ninja from the comfort of their own home now has a way to do so, thanks to Tarlin's new Chiisana Ninja toy collection."

https://soranews24.com/2023/04/20/japansnewtinyninjacapsuletoycollectionturnsyourdeskintoasecrethideout/

The six-piece collection features miniature ninjas going about their business, with a few stylish touches to help us get through the working day.
There's "Kunai" (ninja knife throwing)...
"Kumogakure" (disappearing behind a cloud)...
"Shuriken" (ninja star throwing)...
"Kabenobori" (wall climbing)... "Kakureminojutsu" (Ninja art of camouflage).
Etc...
We can say that the little Ninjas on a mission will be a great way to keep us motivated in our activities as we go about our working day.
The items dedicated to the ninja art of camouflage are sure to bring a smile to our faces every time we look at them. These capsule toys retail for 300 yen, and Tarlin says it is unable to provide precise information on the availability of these items in various stores, so it asks customers to look for the collection at capsule toy vending machines throughout the country.

Apparently, we will have to embark on a Ninja mission to find other Ninjas. We will be part of this small but great world. USB ports, tremble, we are coming!

Now let's proceed into the abyss of perversion. Those who love the world of anime and manga will know very well that in Japan, octopus tentacles have a more or less veiled semantic component related to the erotic sphere. Yes, octopus tentacles.
Remember Lady Oscar's thorny brambles? That scene from the famous theme song did not include tentacles, but those thorny brambles defined the passion and inner torment of a female heir. "But alas, you were born" were the famous words of the Italian theme song.
Similarly, Hokusai's famous painting "The Dream of the Fisherman's Wife," quite explicit for the 1800s, shows two octopuses tightly wrapped around a naked woman, engaged in erotic acts in her mouth and other intimate parts. An octopus on a human being, a strange combination for us Westerners.

https://it.wikipedia.org/wiki/Il_sogno_della_moglie_del_pescatore

From the 1800s to the anime of the 1990s and up to the present day, this "tentacular" ideology continues silently on its path deep within Japanese thought.
A few years ago, a Japanese cosplayer/food blogger/model named Namada titillated the Japanese world by posing for photographer Yamamoto in a photo shoot with a real giant Pacific octopus weighing several kilograms.

https://mothership.sg/2016/09/japanesecosplaymodeldidaseriesofphotosofherselfdressedinliveoctopusjustbecause/

Have you checked out the links? In Italy, does the octopus evoke these fantasies? Could the fisherman's wife in the 1800s have dreamed of anything more exciting?

Who, like me, after seeing that poor 8 kg octopus, thought that the young lady posing with it is certainly pretty, but also thought about the risk of food waste? Do you also think that the art and death of that octopus walk a fine line between regret and questionable communication?

After this long preamble, which is quite incompatible outside Japan, we are all ready to discover the shocking invention of pallets containing plastic tentacles that emerge from a plastic base of a... hypothetical plastic octopus. Colorful emerging tentacles that could wrap around some small "plastic" object placed on top of the base.

Please note that the manufacturer "Proof" has clearly pointed out that there are "many ways to play with" this product and is very proud when "some manufacturers go beyond all limits in the design process" of these objects so densely imbued with meaning.

You can view these scandalous objects on X: PROOForiginal, @proof_original.

In short, we just cannot understand how deeply and intimately the Japanese can experience the concept of tentacles and, above all, of wrapping them around the body.

The creator is keen to emphasize that with this "tentacular" product, we can wrap all the characters in our display cases, creating "very scandalous" positions.

What's more, this product is just a small object that introduces us to the wider world of "Tentacle Paradise" which... from 2019 to today, 2025... has not attracted much media attention... sorry...

Prices: four coins for a few "plastic" tentacle strands.

Tentacles in multiple colors.

Tentacles with suction cups.

Tentacles with small protuberances.

Smooth tentacles.

Stop. Nothing else to add.

I could say that the word "Tentacolare" in Italy brings to mind the beloved film "Johnny Mox" by the beloved Roberto Benigni, and perhaps we have already forgotten both the lovely Miss Namada and the colors/shapes of the tentacles.

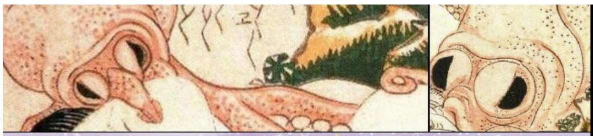
Well, let's relax for a moment.

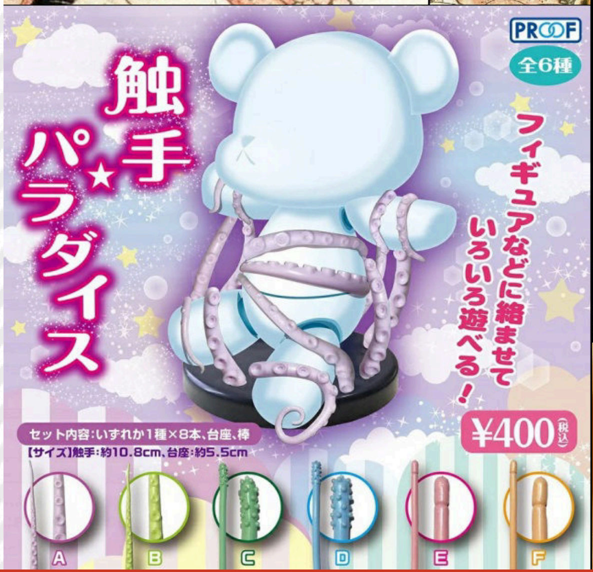
Let's talk about a fairly innocent little ball, a dispenser of ordinary shopping bags, but nicer than normal plastic wrappings.

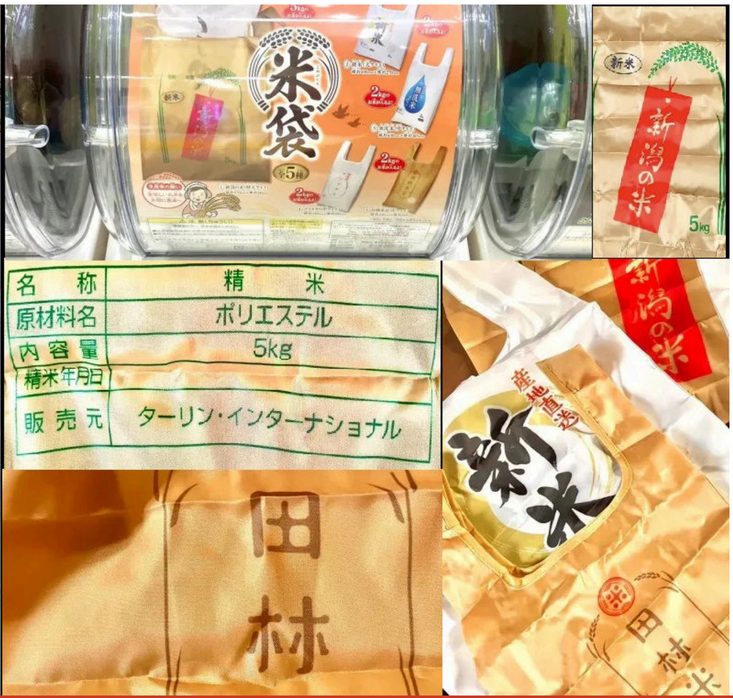
We're talking about bags that contain rice for meals. Size 1:1, so they are not miniaturized bags sold inside Gachapon balls. Manufacturer: Tarlin.

https://soranews24.com/2023/12/12/gachacapsuletoymachinesellsjapanesericebags/

Reporter Udonko in Niigata reviewed this "new way of shopping at the supermarket."




触手パラダイス
PROOF
全6種
フィギュアなどに絡ませていろいろ遊べる!
¥400(税込)
セット内容:いずれか1種×8本、台座、棒
【サイズ】触手:約10.8cm、台座:約5.5cm
A
B
C
D
E
F



These polyester bags, carefully folded to fit into the pallet, are rich in detail, indicating what type of rice they should carry, how many kilograms they can correctly hold up to the fill level, the date of processing, and various other data, including the name of the rice seller, which in this case coincidentally is "Tarlin," the manufacturer of this type of pallet.

After this moment of relaxation, let's return to something that will disturb your sleep. Who has seen the entertaining animated film Pompoko? The film told the story of a charming family of tanuki, or raccoon dogs. This delightful and amusing community of animals had to survive human urbanization, which was dangerously encroaching on their forest: their home. It is a highly entertaining film that raises awareness of the difficult coexistence of humans and nature.

Well, these raccoons had a fun means of flying transportation, not an airplane, not a catapult, not a flying carpet made of cloth, but a part of their own body stretched and elongated and pulled out of proportion until it became as spacious and flat as a carpet.

Which part of their body, obviously genital, am I talking about?

I'm not joking, the painter Yoshitoshi was already painting these Tanuki in 1890 with extremely stretched scrotal skin... like a ship's sail!

A gust of wind and off they go: flying raccoons!

Here's what to put in the gachapon balls: a humorous version of stress balls called "Squeezable tanuki testicle."

Manufacturer? Tarlin, as always.

This company is brilliant. If it didn't exist, it would have to be created.

Price? Four coins, 400 yen.

Gentlemen, we must be serious people, the price does not refer to a... obviously it refers to the natural pair of balls physiologically attached to the same Tanuki!

Are you thinking of buying the ball/balls?

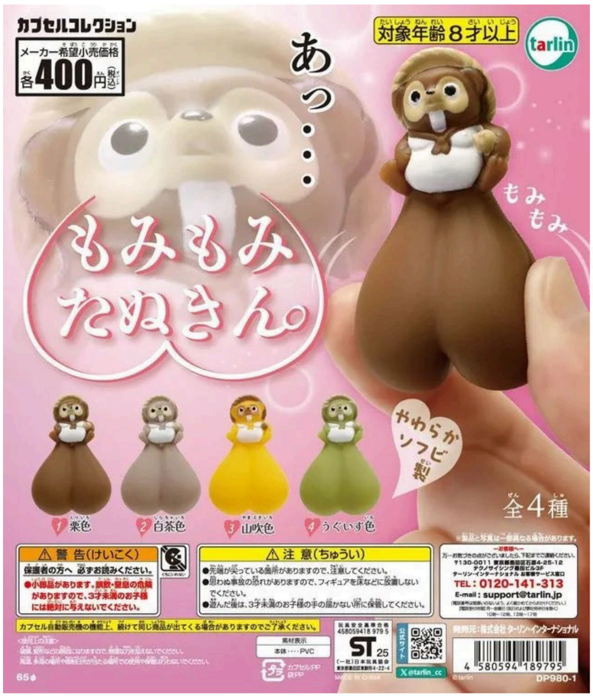
I hope so, me too!

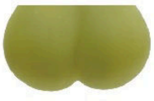
What's more, this product comes in kuri, shirocha, uguisu, and yamabuki colors. Obviously, the most relevant color is... find out for yourself by looking up the meaning of these elegant and high-sounding names.

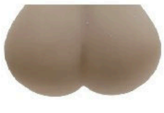
One last anecdote: note that Tarlin International states that "These products are intended for users aged 8 and up. When not being used for play, the characters must be stored in a safe place out of the reach of children under the age of 3, as a safety precaution and also to spare us the embarrassing conversation of having to call an ambulance and scream: Help! My son is choking on Tanuki testicles!'"

These folks at Tarlin are truly adorable and brilliant. Thank you so much for your ingenuity!

That's all for now, friends. Thank you for reading the article. I invite you to read the next one, where we will add a Gacha to our... Pon!







The unknown story of Takeru, the Japanese ancestor of Steam

by Takahiro Yoshioka – Translation by Carlo Nithaiah Del Mar Pirazzini

In the mid-1980s, when the world was discovering the first mobile phones and Nintendo's Famicom was taking its first steps, Japan was already experimenting with a revolutionary innovation that heralded the era of digital distribution. This "leap forward" was called Takeru and represented the first serious attempt to create a dematerialized software sales system.

Brother Industries and the birth of a pioneering concept

The story begins at Brother Industries, a Japanese company that had established itself in the production of sewing machines before diversifying into typewriters and printing solutions. The company introduced the Takeru system in 1985 with 15 test machines, followed by a commercial launch in 1986. Faced with the proliferation of video game shelves and the logistical problems that came with them, Brother was looking for an innovative solution to manage inventory and avoid stockouts.

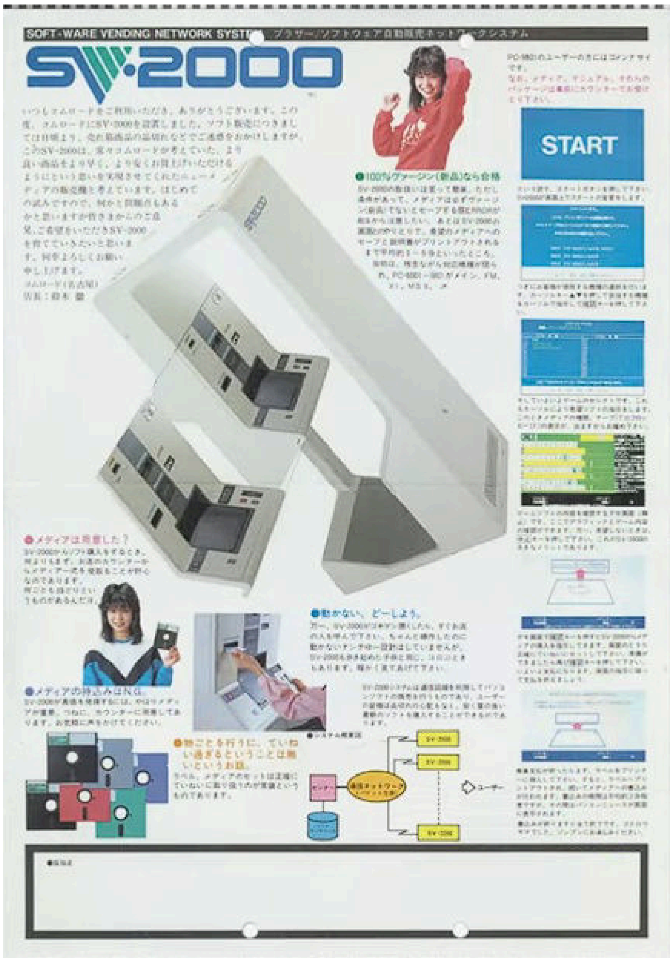
The idea was born out of frustration with a Japanese software market so dynamic that retailers couldn't keep up with demand. Some titles flew off the shelves as soon as they arrived, while others piled up without finding buyers. Takeru, named after Yamato Takeru, a legendary Japanese hero, offered a radically different approach.

A revolutionary vending machine

The first commercial version, called the SV-2000, worked on a simple but revolutionary principle for the time. Users selected a game or software on a touchscreen, paid cash, and then inserted a rewritable MSX cartridge into the machine. The system then downloaded the content via a dedicated line and printed out an information leaflet.

This first generation, however, had significant limitations. Each game took up to 20 minutes to download over the telephone lines of the time, which were limited to about 1.2 kilobytes per second. This slowness made the experience frustrating and drastically limited the number of sales possible during store opening hours.

Technological advances marked the arrival of the SV-2100 in 1986, which introduced 3.5-inch floppy disks and equipped each machine with a hard drive to store the most popular titles. These improvements significantly reduced waiting times, from twenty minutes for a cartridge to a few minutes for a floppy disk.


SOFT-WARE VENDING NETWORK SYSTEM
SV-2000
START

The SV-2000 in all its advertising glory!

The golden age of Takeru

The third generation, the SV-2300, launched in 1991, represented the pinnacle of the system. This version was one of the world's first commercial applications of ISDN, offering speeds ten times faster than the previous model. Equipped with a laser printer and 5.25-inch floppy disks, it abandoned cartridges for good.

Takeru was not limited to the MSX standard. It supported an impressive variety of platforms: PC-8801, PC-9801, X1, FM-77AV, FM Towns, IBM DOS, Windows, and even Macintosh. At its peak, the network had 268 machines distributed throughout Japan.

The introduction of the Takeru Club added a social

メンバーで作るコミュニケーションスペース
創刊準備号
TAKERUわあるど
World
TAKERUキャラクターの
ニックネーム大募集!!
創刊号にて発表！
5名様に編集部製作オリジナルTシャツをプレゼント！
ぼくに名前をつけてクレヨン
TAKERU
募集してま〜す！
その1 楽しいおたより
その2 個性キラリのイラスト
その3 笑える4コママンガ
その4 何でもQUESTION
その5 ソーサリアンMSX2移植記念イラスト募集
その6 力作来い来いプログラムコンテスト
その7 活躍してもらっちゃう TAKERU CLUB特派員20名大募集！
TAKERU事務局よりごあいさつ

Takeru World in 1991

dimension to the service. Members received discounts, detailed manuals, and subscriptions to the quarterly magazine Takeru World. These publications also served as an advertising showcase, listing available games and the location of the machines.

The indie revolution and exclusives

One of Takeru's most notable innovations was its openness to the amateur and independent world. Brother partnered with independent developers to offer hundreds of games distributed exclusively through Takeru. This approach anticipated the spirit of modern platforms such as Steam or itch.io, giving visibility to independent creators.

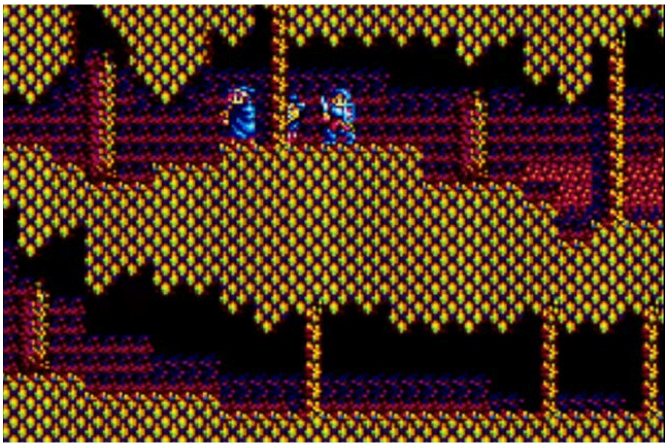
However, Brother maintained strict quality control: on average, two out of three games were rejected. This selection is reminiscent of current app store policies, demonstrating a forward-thinking approach on the part of the Japanese company.

For traditional publishers, Takeru represented an interesting opportunity to distribute additional content, mods, or old titles. Among these were the famous Doom patches. Brother received a 30% commission on each sale and paid 70% to the developers, a business model that would later be adopted by Apple and its competitors.

The decline and legacy of Takeru

Despite its innovative nature, Takeru was unable to withstand the technological changes of the mid-1990s. The rise of the web, the advent of shareware, and the spread of CD-ROMs radically transformed software distribution. The revolution was underway, but Brother was already a few years ahead of its time.

The system was officially discontinued in February 1997, but its legacy lived on in an unexpected form. The team behind Takeru turned their attention to something that would redefine entertainment for generations to come: karaoke. The technologies developed for Takeru found a second life in Joysound karaoke, revolutionizing the industry by enabling the distribution of music tracks over the internet.



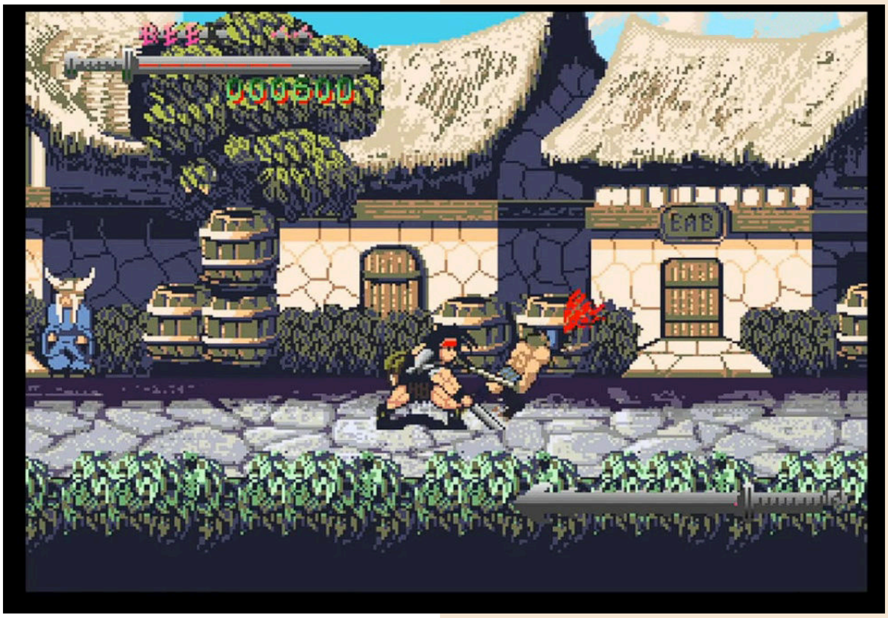
SORCERIAN PC game 8801 by Nihon Falcom, distributed with Takeru

The Takeru Machine at work

NEW GAME

KROGHARR

Year: 2025
Editor/Developer: Tigerskunk
Genre: Beat’em up
Platform: Amiga
Website: https://tigerskunk.itch.io/krogharr



"Come on, my scrotums!!! Clench your crabs, my Sbabbari... We will pass through desolate places, deserted beaches, wild forests, mosses, fungi, and lichens..."

Never touch a barbarian's beer! Never even think of stealing it from the hometown of a video game protagonist! Never! It's madness... it's heresy... No! It's Krogharr!

The terrible chieftain Pilsator thought it would be a good idea to do just that, unleashing the wrath of Krogharr, the berserker hero and, as the title suggests, the protagonist of the game. What will all this desire for beer lead to? Well, healthy and powerful barbaric "ultraviolence" complete with pixelated smashing, mauling, and decapitations.

Krogharr is Tigerskunk's latest Amiga effort. A title that has been a long time coming but which we can now enjoy while savoring a good beer in front of our joystick. Ready to fight.

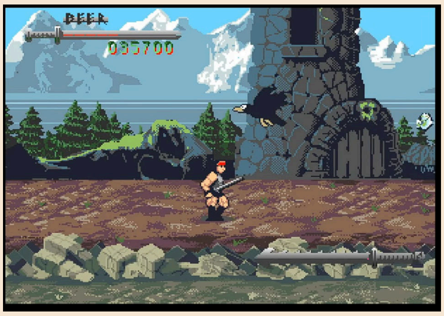
There's a bit of Barbarian, a pinch of Golden Axe, and a dash of Tiger Road in this title. The goal is to recover the beer, and you do so by shredding hordes upon hordes of filthy creatures that will come at us as we wander.

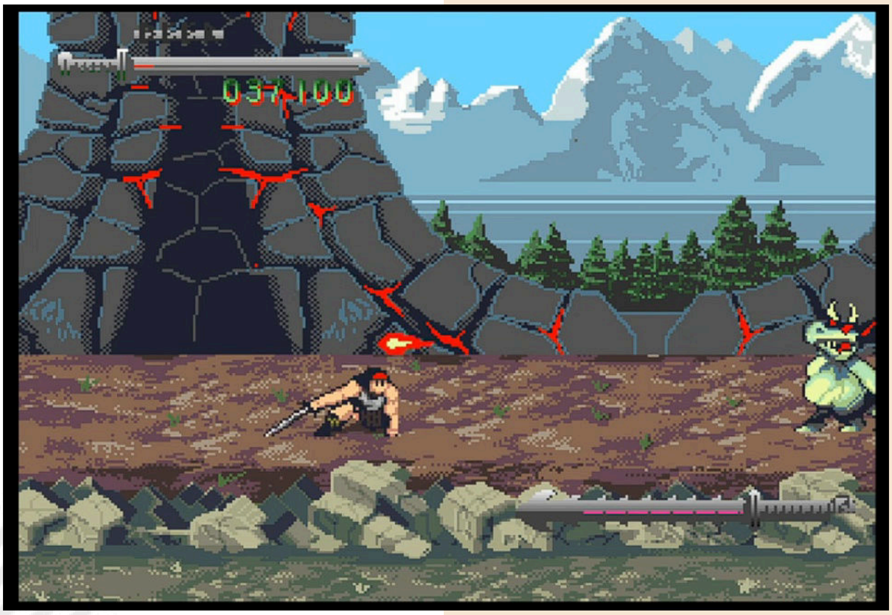
At the end of each level is the classic boss. Usually a burly rogue in bizarre clothing (see the Tyrolean ogre) but with a desire to kill us.

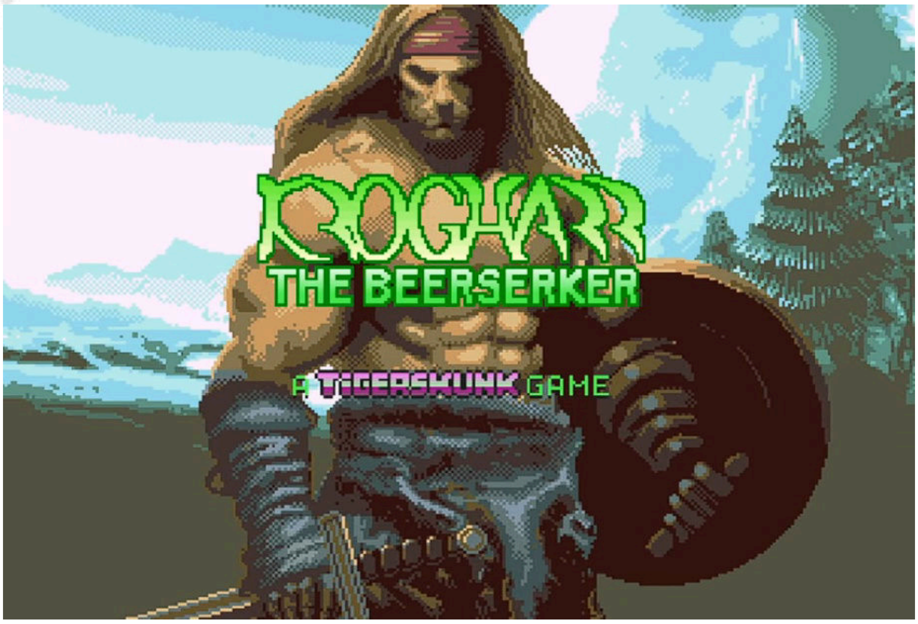
Krogharr will slash left and right and, thanks to the beer, will be able to go berserk for a short (but substantial) period of time, showing off his ability to chop off heads on the fly.

The game is well-made and fun, perhaps a tad repetitive in the various enemy waves. But in the end, all beat 'em ups of this genre are.

The graphics are very polished and colorful. There is a style that reminded me a lot of some Psygnosis titles. Personally, I really appreciate the color choices used.







OUR FINAL SCORE

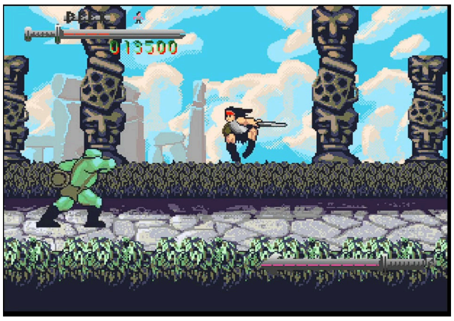
» Gameplay 85%

Simple... actually, super simple! You move forward by slashing everything in your path. No need to think too much, just have fun.

» Longevity 80%

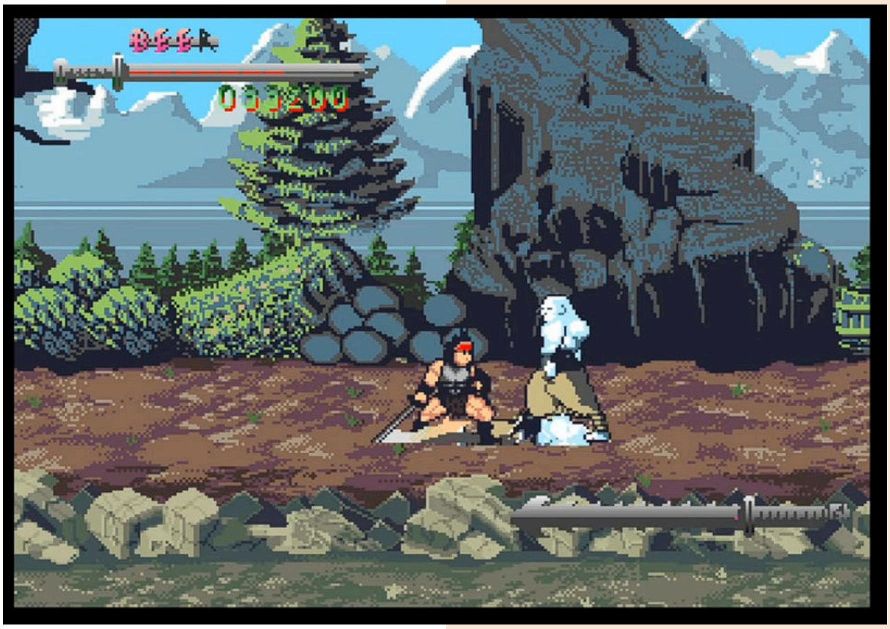
It may not be very varied, but it's definitely challenging.

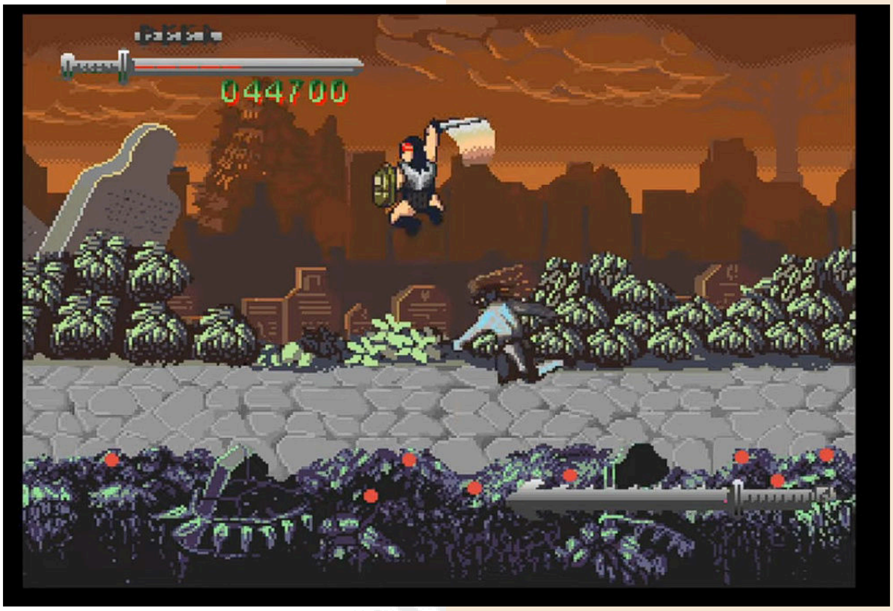
The title is not innovative, it doesn't have any crazy gameplay... but it's fun! Damn fun and definitely gory. Just like the good old games of yesteryear, where all you had to do to be happy was blast waves of enemies of all kinds.



Krogharr does just that. It's simple, well made, colorful, and definitely fun. You can play it on any Amiga with 1MB of total RAM, in emulation, and on CD32 (in ISO format).

by **Carlo Nithaiah Del Mar Pirazzini**





NEW GAME

OUTRUN AMIGA EDITION

Year: 2025
Editor/Developer: Reassembler
Genre: Driving game
Platform: Amiga AGA – PAL, 2Mb Chip, 8Mb di Ram, 68030
Website: https://reassembler68k.itch.io/outrun-amiga-edition

It took 39 years, but we finally got a worthy port of OutRun on Amiga.

Like many popular arcade games of the 1980s, Sega's OutRun was adapted for numerous home platforms: Mega Drive, Master System, PC Engine, ZX Spectrum, C64, and many others.

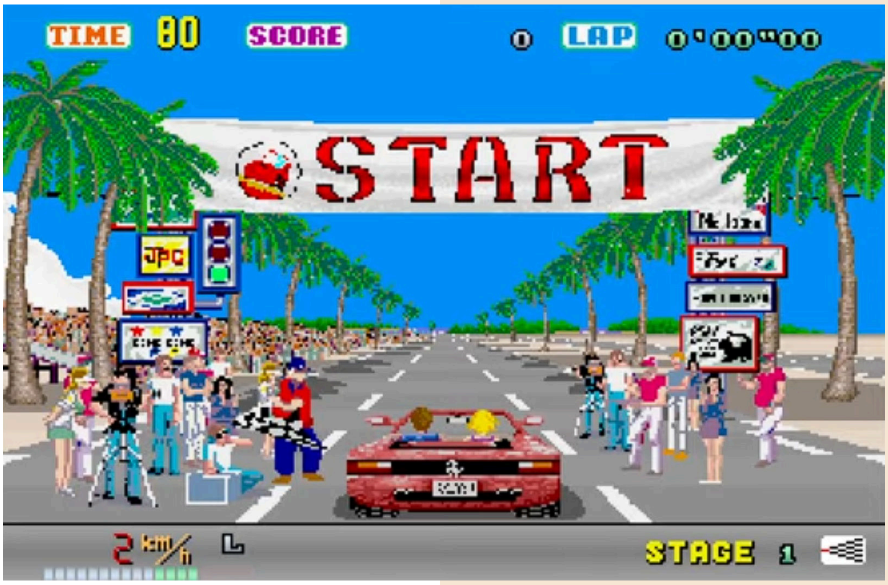
In 1989, it was also converted for the Amiga, a system that was more than capable of handling a decent conversion, but which saw the arrival of the US GOLD port, which went down in history as one of the worst conversions for the Amiga ever. Apparently, it was a direct port from the ST, which meant that the Amiga's powerful blitter could not be used.

It lacked everything. Slow, technically sloppy, and above all, completely lacking in the feeling that the arcade game gave... freedom!

Yes, because OutRun gave you the freedom to travel. There were no monsters to defeat or obstacles to overcome. There was a long track to conquer, a beautiful blonde by your side, and a convertible Ferrari Testarossa.

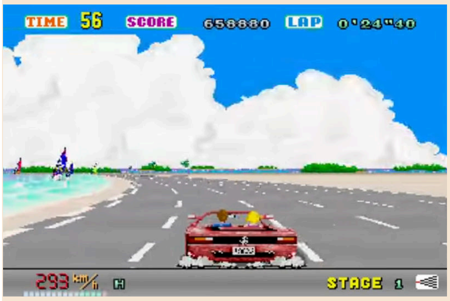
After all this time, Reassembler has rewritten everything and come up with an (almost) perfect port. Demanding in terms of configuration, but definitely more akin to the arcade game.

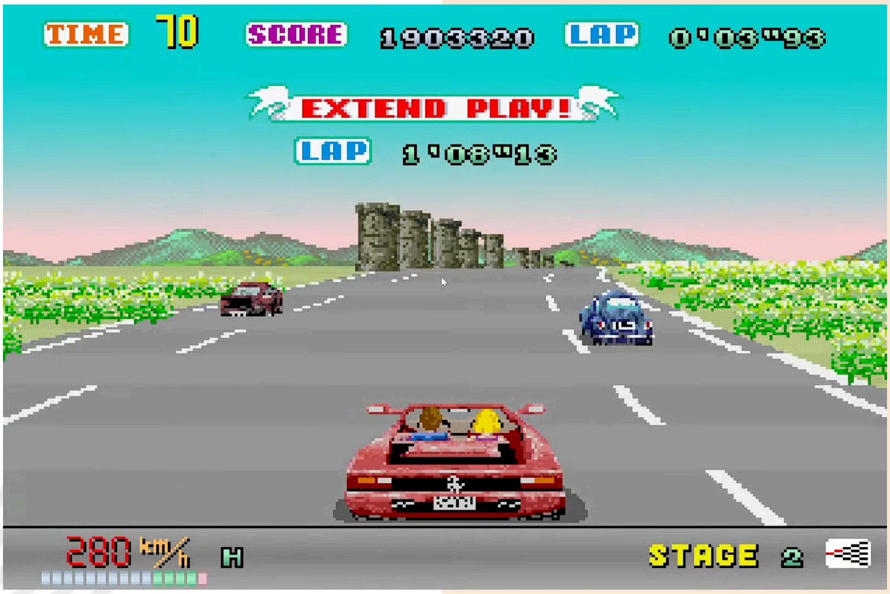
The game can be downloaded for free from the developer's itch.io page: it contains the executable, a Python script, and the arcade audio in Amiga format rewritten with 4-channel Protracker MOD.

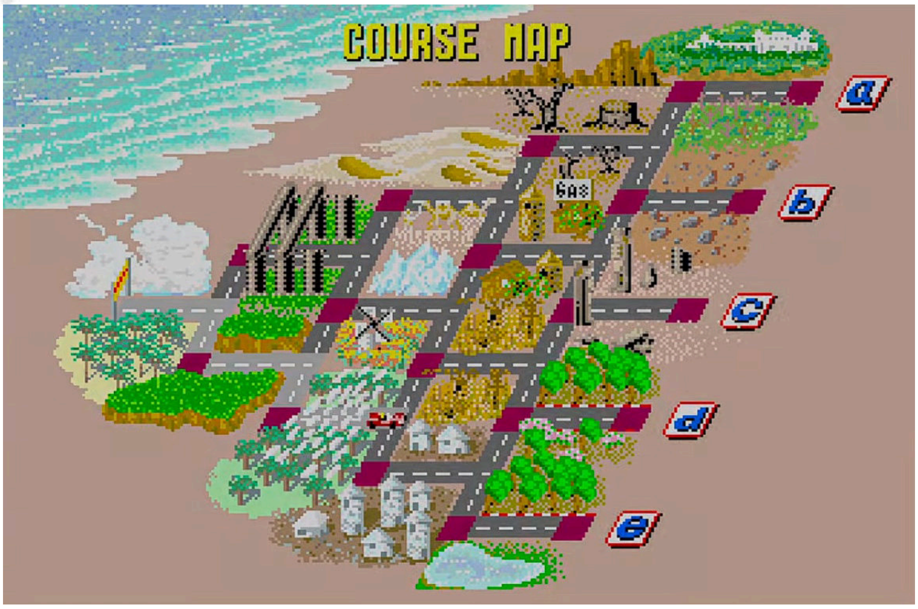


The ROM data required by the game is not included. In order to work, it requires some original arcade ROMs. These must be converted using the Python script and placed in the game folder.

The game is there, with some technical compromises compared to the arcade,







OUR FINAL SCORE

» Gameplay 85%

Everything is there. Graphics, sound, and gameplay. Requires high-end configurations and a bit of "work" on the ROM conversion.

» Longevity 90%

Keeps you glued to the screen like in an arcade. One game after another.

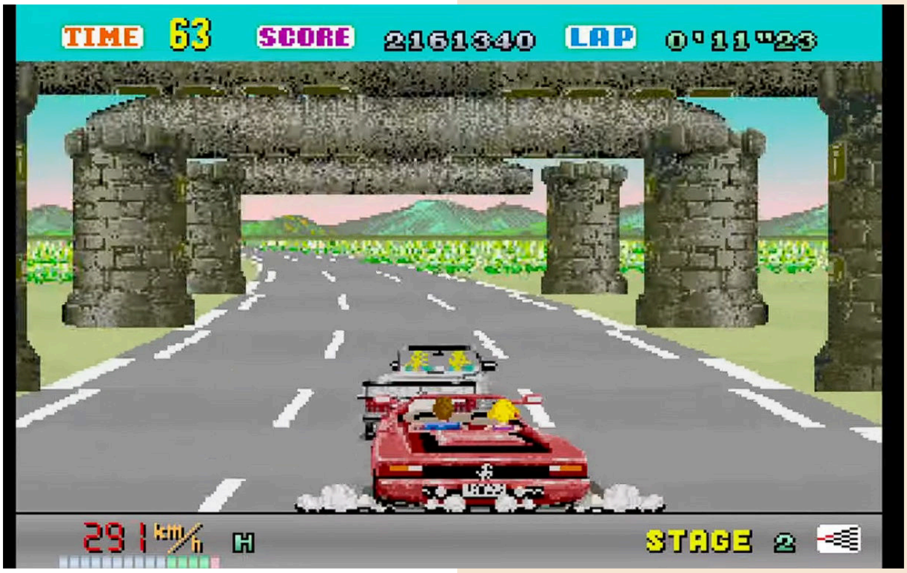
but it's there. The paths are there, and so is that feeling of freedom that was missing in the US GOLD title.

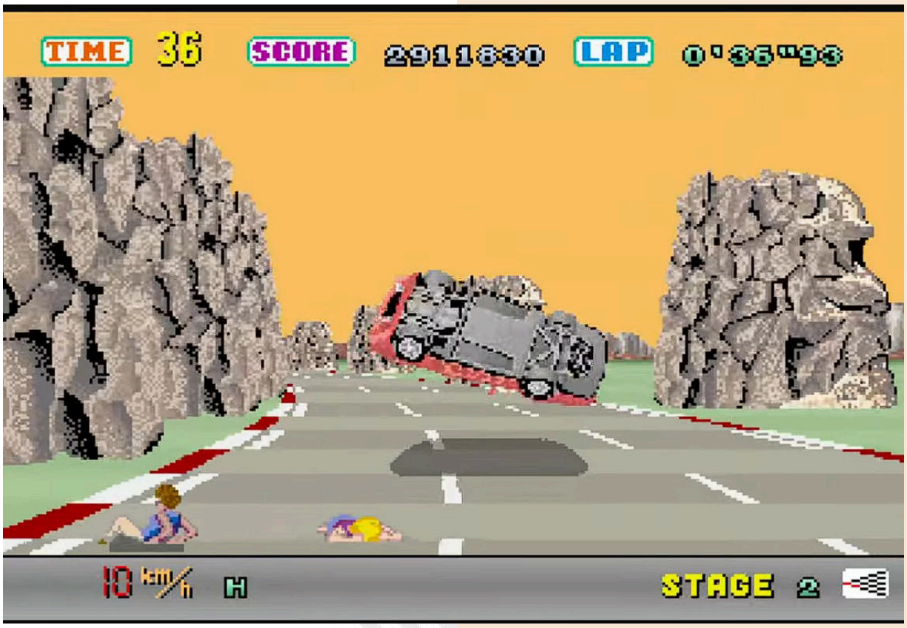
The technical requirements are quite high. On a standard A1200, the game runs but is very slow and jerky. More demanding hardware upgrades are recommended (all specified and tested by the developer and listed on the page). I am convinced that in 2025 this type of problem will no longer be as serious as in the past. Amiga users have 'evolved' by buying peripherals and new hardware, or by installing emulators such as WinUAE that allow them to fully enjoy titles like this one.

Graphically, it's OutRun. Even after 39 years, the aesthetics are very pleasing. The original soundtracks are also iconic and very well recreated.

The gameplay of the time is very much present. That mixture of enjoyment mixed with the relentless coin-eating system. A lethal but very effective mix. A fast, fun, and well-made game that we can't help but recommend.

by **Carlo Nithaiah Del Mar Pirazzini**



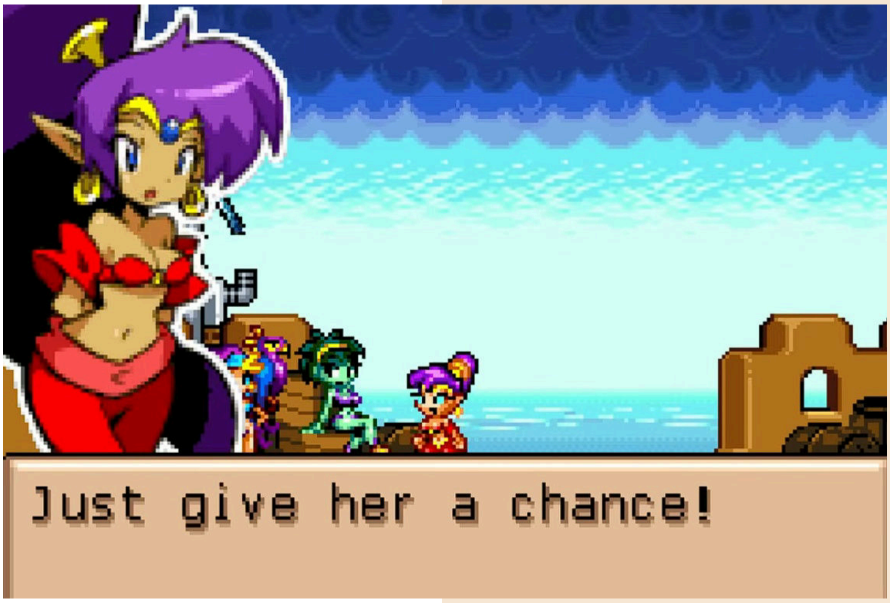


NEW GAME

SHANTAE ADVANCE: RISKY REVOLUTION

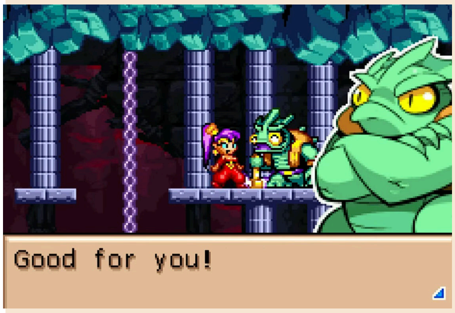
Year: 2002/2025
Editor/Developer: Limited Run, Way Forward
Genre: Platform/Metroidvania
Platform: Game Boy Advance
Website: https://limitedrungames.com/products/shantae-advance-risky-revolution-gba#product-details

The "lost chapter" has come to us! Finally, one of the most anticipated platform games of the early 2000s sees the light of day on our Game Boy Advance, shelved for more than 20 years and finally available thanks to the skilled work of the developer and the guys at Limited Run.
A funny and very colorful saga that had the misfortune of never seeing the light of day, with this episode, due to the lack of a publisher. Our protagonist is a "half-Djiin," a genie of the lamp who finds herself having to stop the evil plans of her longtime rival, Risky Boots.

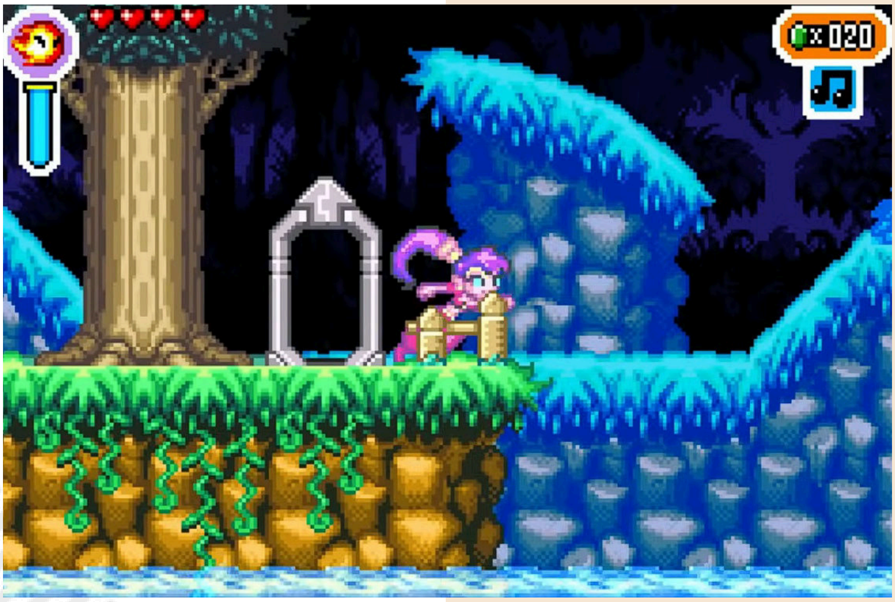

Just give her a chance!

The game is reminiscent of Wonder Boy in Monster Trap, both in terms of gameplay mechanics and settings. The most interesting part is undoubtedly the mechanics of moving between floors. In different levels of the game, you can move from one floor to another in order to continue the adventure. The execution is very reminiscent of Kirby Triple Deluxe. This mechanic increases the "puzzle" aspect of exploring the levels and is enjoyable.

There is no game map, which is definitely a downside. Without a map, it is easy to get lost between the different worlds, even though the text on the screen will "remind" you several times where to go and what to do. A map would have been very useful in exploring the many dungeons.
The graphics are stunning. The Shantae team has clearly pushed the GBA hardware to its limits to bring out the beauty of the game worlds. The use of color is masterful, as are the animations and the variety of sprites in the game. The effects of the "rotating" levels are also beautiful, a style that reminded me of some of the game worlds in the Castlevania series on SNES.


Good for you!

But the story is definitely the game's strong point. The entire Shantae saga is based on cheeky and entertaining storytelling, and Shantae Advance



OUR FINAL SCORE

» Gameplay 90%

Fun, simple, with a beautiful narrative, first-rate technical aspects, and an interesting gameplay system that mixes platforming with Metroidvania.

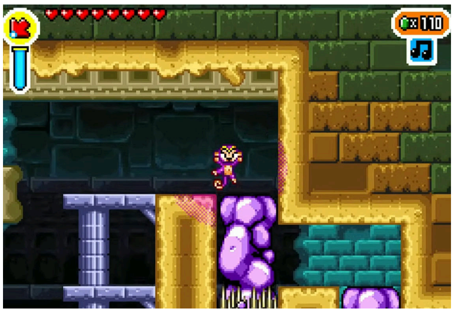
» Longevity 85%

The MAP!!! All they needed to do was add an effective mapping system and Shantae would have been perfect!

does not disappoint. It's full of jokes, and the dialogues are hilarious and entertaining. It's enjoyable to listen to the crazy speeches of the protagonists or the "delusions of omnipotence" of Risky Boots.

I'll get straight to the point at the end of this review. I can only recommend purchasing this title for GBA. On the Limited Run website, you can buy the beautiful cartridge with manual, artwork, and mini-poster. It's expensive but definitely an important piece for collectors. The game is also available for modern platforms on STEAM or in dedicated digital stores.

by **Roberto Del Mar Pirazzini**



Ape with Kitten - artist unknown

NEW GAME

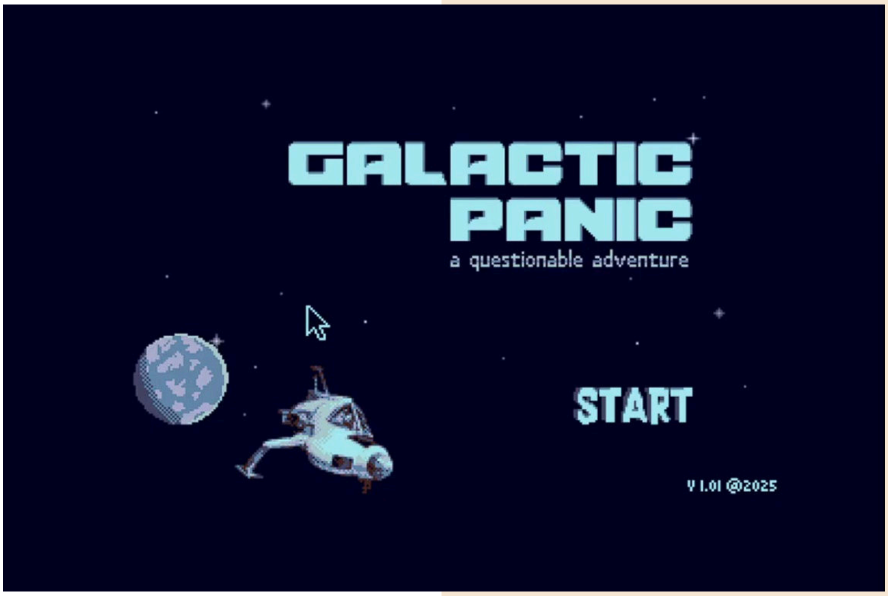
GALACTIC PANIC

Year:2025
Editor/Developer: Insert Studio Name
Genre: Avventura punta e clicca
Platform: Atari ST
Website: https://tinyurl.com/yeyha2bn

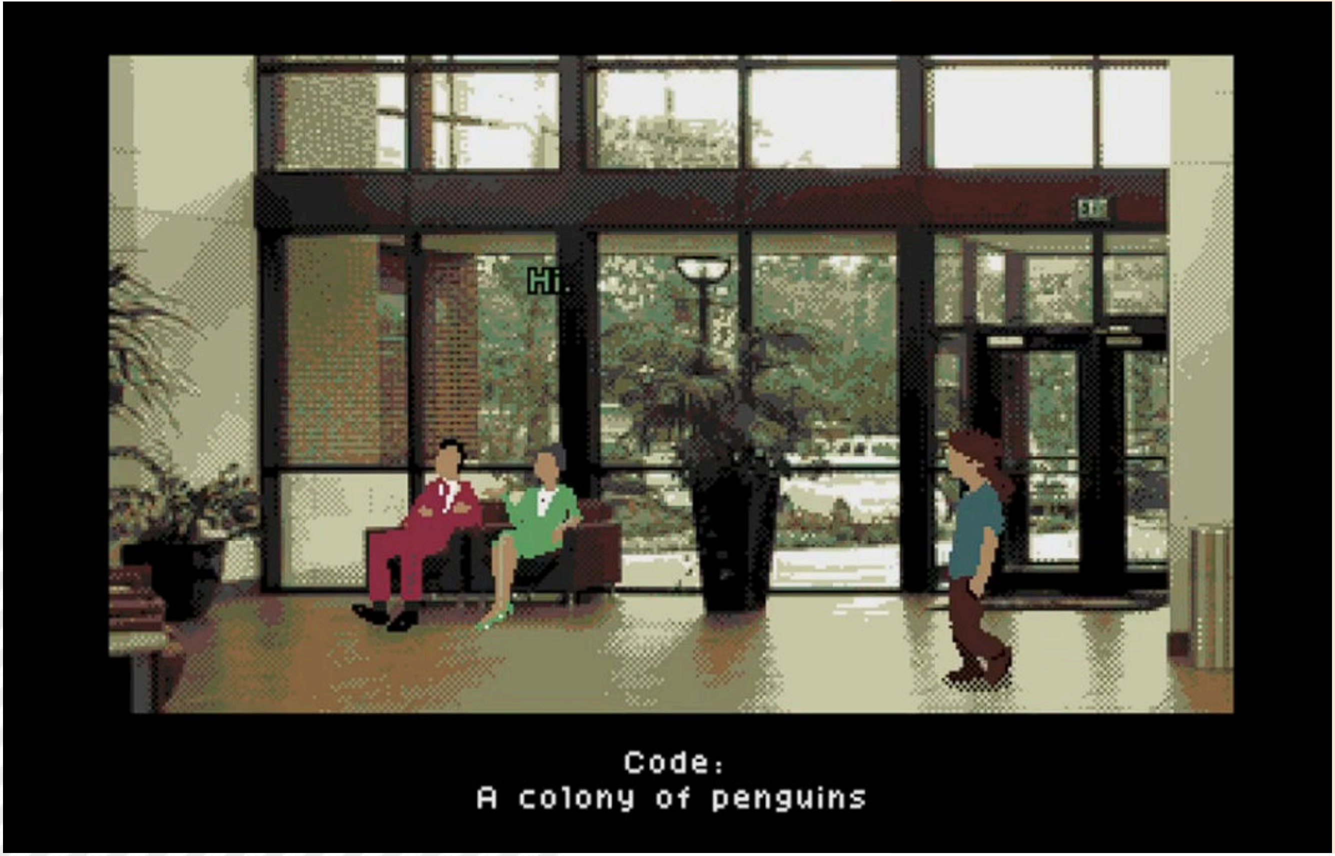
"A rather silly point-and-click adventure in 16 colors." This is how the game is introduced on the developers' itch.io website, and Galactic Panic is indeed just that. Ironic, funny, and crazy.

It is reminiscent of the style of LucasArts' graphic adventures from the 1990s.

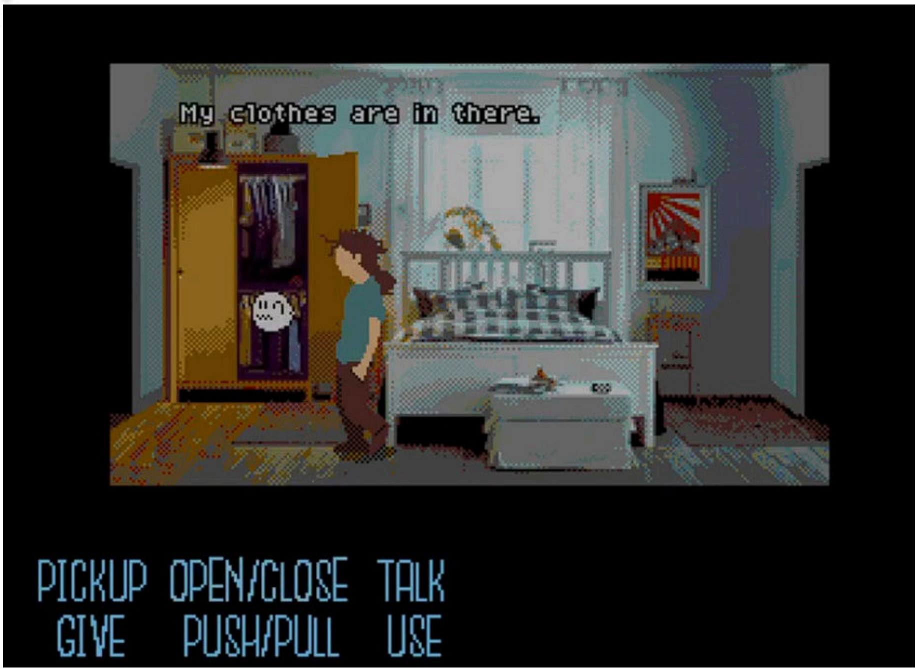
The protagonist of the game has been recruited to embark on a dangerous interstellar journey. The inhabitants of the planet Zirius 42 are facing a terrible threat and need our hero. It must be admitted that the inhabitants of the planet in question are not exactly the sharpest tools in the shed, and at a certain point, the adventure takes a... peculiar turn!



The title features a LucasArts-style gameplay system with a simple action selection menu. The plot flows madly and happily across a hundred screens, with no deaths and two game



Code:
A colony of penguins



OUR FINAL SCORE

» Gameplay 95%
An "old school LucasArts" title with beautiful graphics, a great gameplay system, and lots of irony. MINE!

» Longevity 85%
Not too long, but very enjoyable.

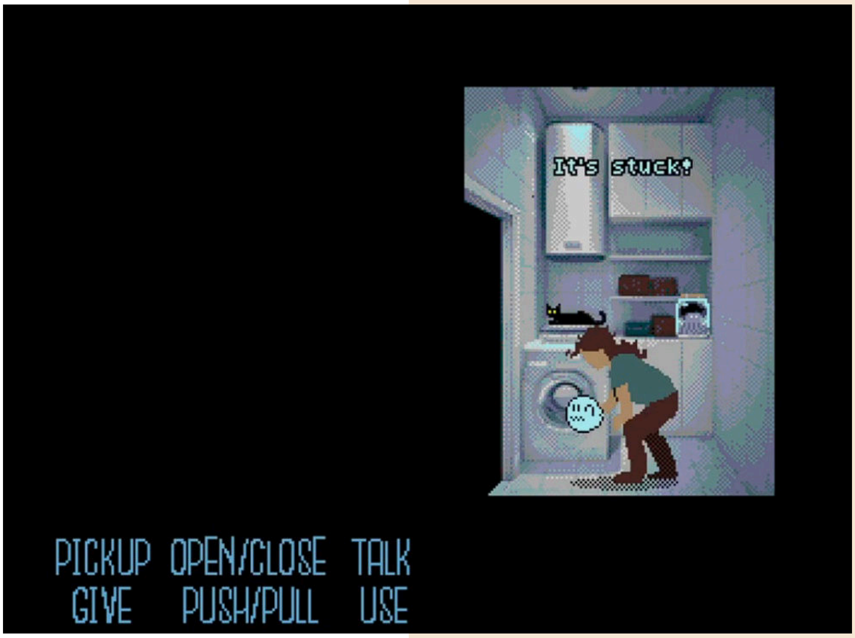
languages to choose from (English and French).

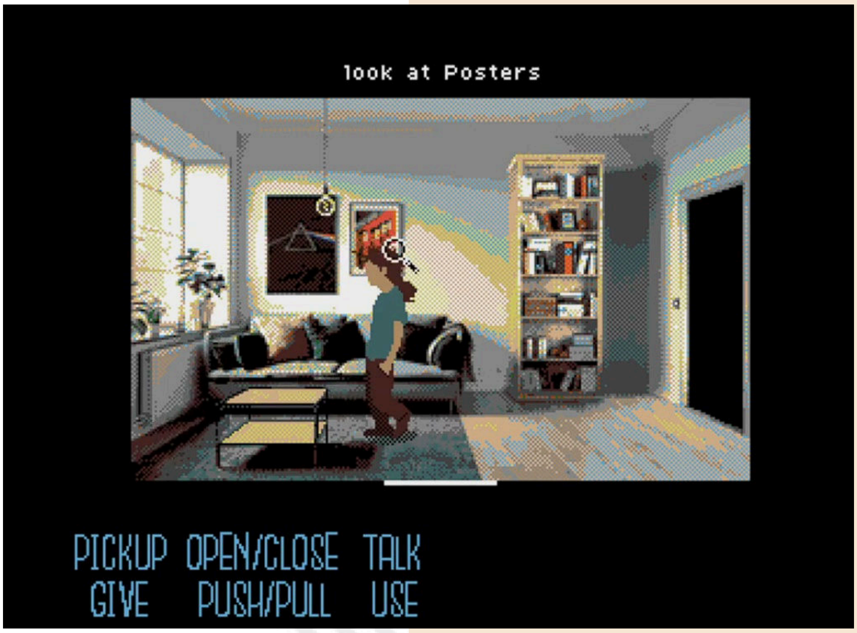
The 16-color graphics work very well, mixing digitized images with pixelated graphics.

If you have an STE or higher system, the game will also feature digitized sound effects. The minimum requirements to run the game are an 8Mhz ST with 512kb and a hard drive for installation. The developers' page also has files for playing in emulation and a nice "ready" file for Windows, where you can just click and play without worrying too much about emulation configurations.

As for the game itself, we can say that the dialogues are hilarious and irony is the glue that holds our entire mission together. Laughter is the purpose of the story, and I assure you that Galactic Panic will make you laugh heartily.
It's beautiful and fun, and it's for ST. Enjoy it.

by **Giampaolo Moraschi**





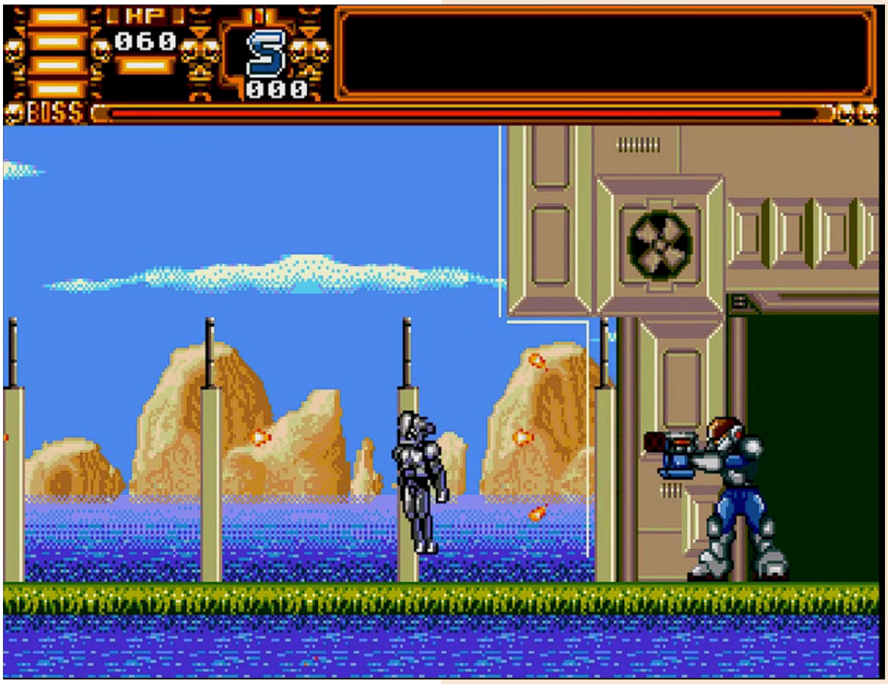
NEW GAME

THE CURSED LEGACY

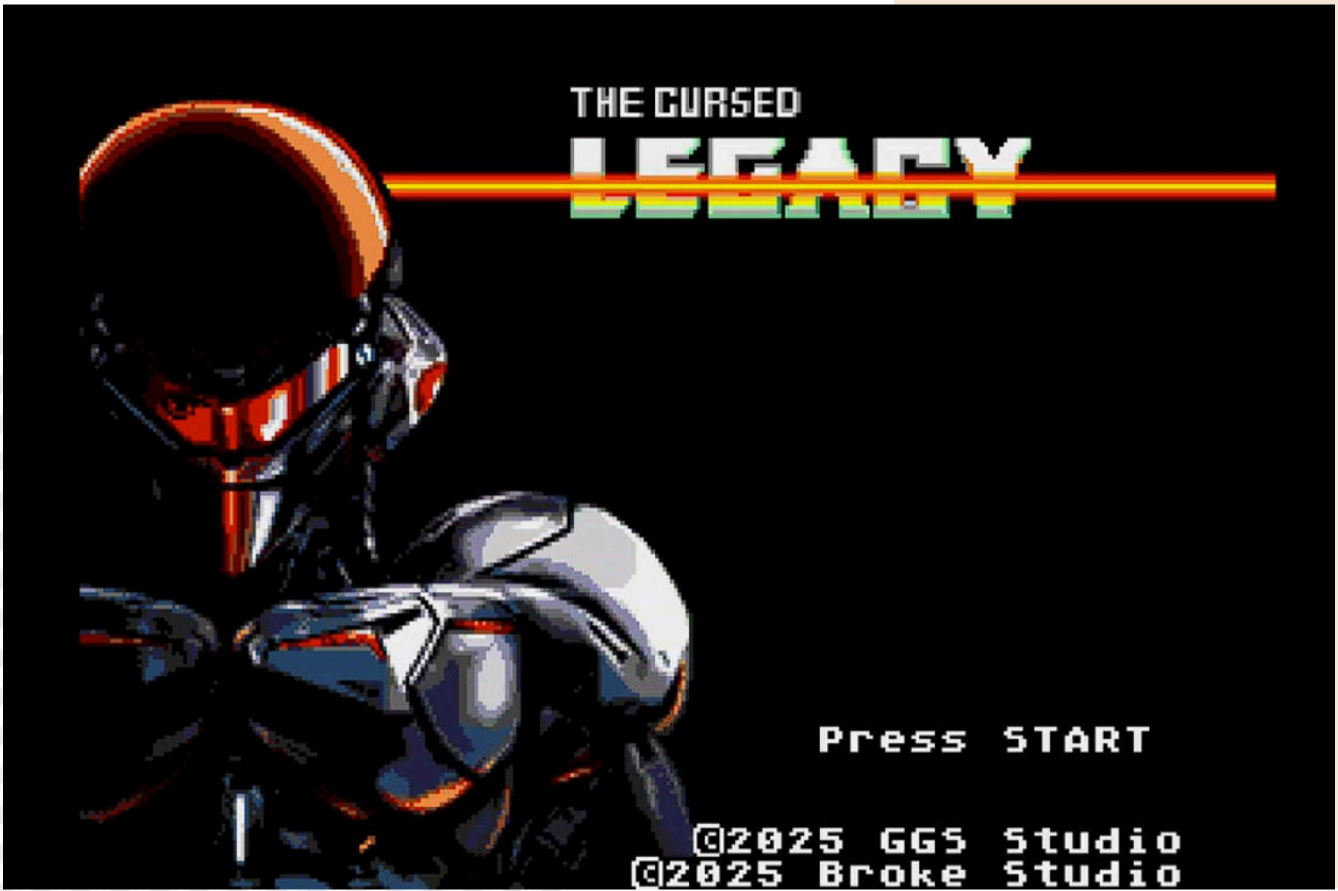
Year:2025
Editor/Developer: Broke Studio, GGS Studio.
Genre: Run n' Gun/ Metroidvania
Platform: Sega Mega Drive
Website: https://broke-studio.itch.io/the-cursed-legacy

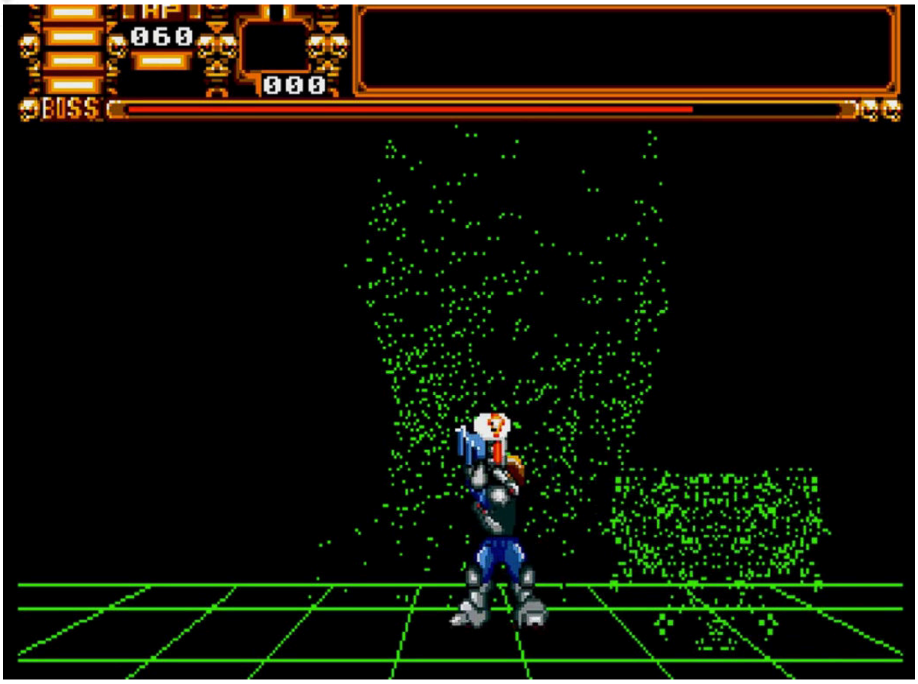
The last GSS Studio title I played was the excellent The Cursed Knight from 1922 (reviewed in issue 43 of 2023), also distributed by Broke Studio.

This new title also arrives on Mega Drive with the same production quality and compelling retro gameplay that mixes two almost opposite genres: hard-core run 'n gun and the sophisticated style of Metroidvania (and there are also several mini-games).



In the ruined universe of The Cursed Legacy, we take on the role of a young mercenary named Leif, initially hired to steal a hyper-technological combat mech, but who sees the trajectory of his mission change over the course





of the narrative. Divided into three sections, the game begins with a shoot-and-run phase before branching out into a more exploratory Metroidvania-style map with a decidedly different look.

During the exploration phase, Leif will be involved in some classic mini-games (special versions of snake or minesweeper) that affect the unfolding of the adventure. A nice innovative touch that is particularly appreciated. As in most arcade titles, there are also some "tough" and substantial Boss Fights that require a lot of patience and memory (you need to understand the bosses' behavior in order to destroy them).

The technical aspect is very well done, as in other GSS productions. The graphics work very well in the more arcade-like sections, while in the Metroidvania sections they are well reproduced but with a slightly 'simpler' color palette.

The audio is top-notch, with a plethora of sound effects of various kinds and ambient sound that suits the atmosphere of the game.

All in all, The Cursed Legacy is a solid action game with generous level design and an almost perfect fusion of game genres. A title to consider in all available versions (digital, physical, or on Steam).

by **Carlo Nithaiah Del Mar Pirazzini**

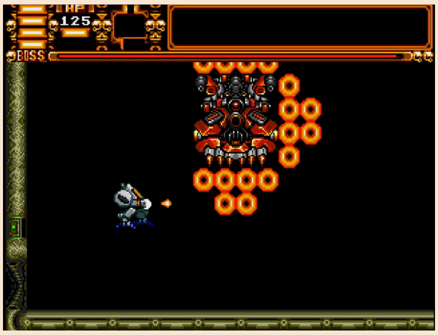
OUR FINAL SCORE

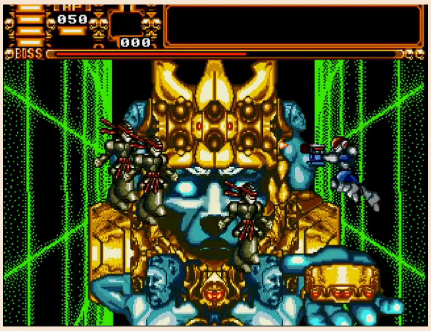
» Gameplay 90%

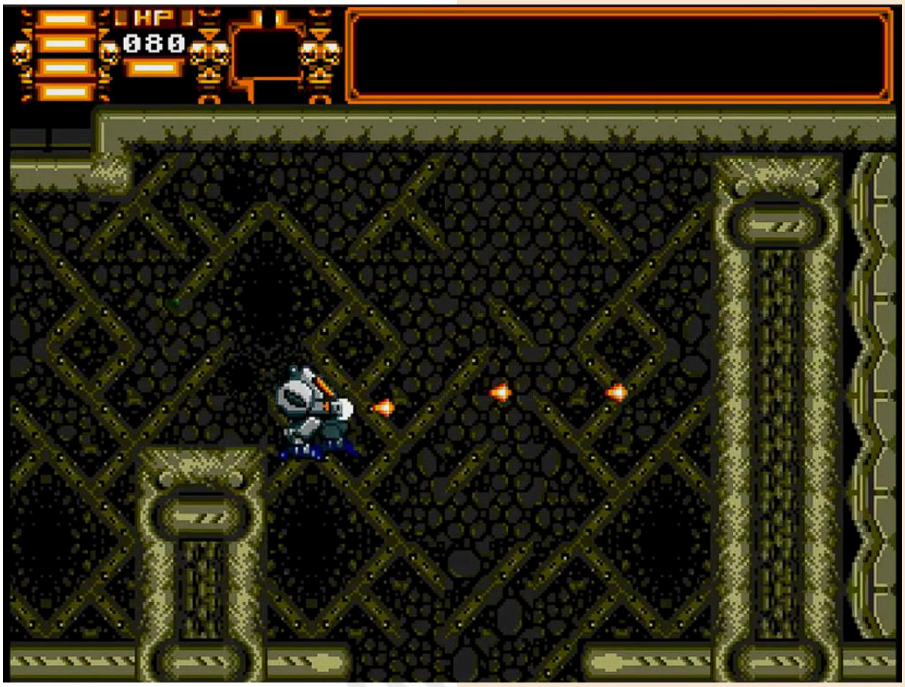
The mix of action and Metroidvania is almost perfect. The gameplay keeps you hooked, and the boss battles are incredible.

» Longevity 90%

Very well balanced in terms of difficulty and level design. A gem among new productions.







NEW GAME

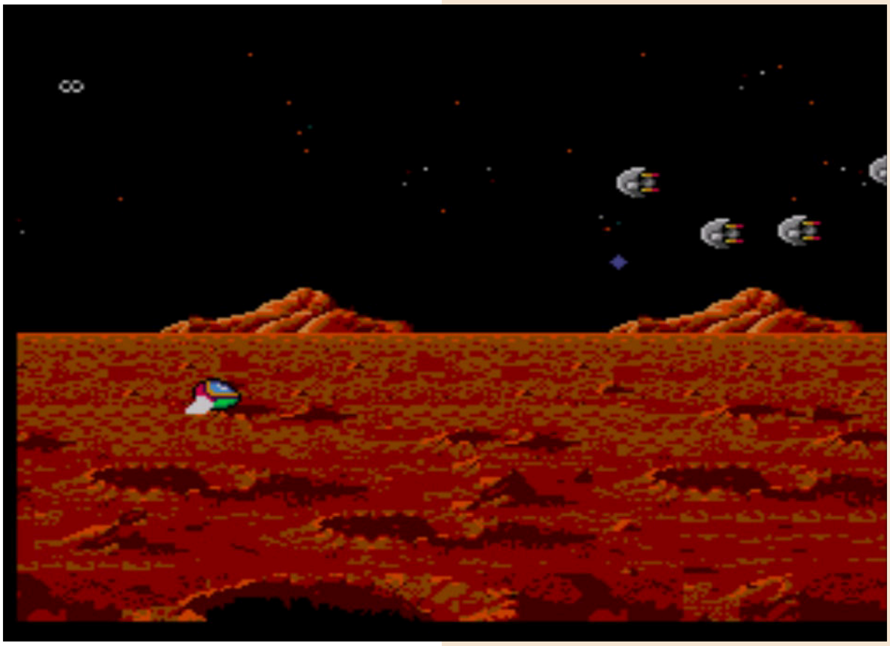
LUNAR SKIRMISH

Year: 2025
Editor/Developer: Eruiz00
Genre: Shoot'em up
Platform: Sega Master System
Website: https://www.smspower.org/Homebrew/LunarSkirmish-SMS

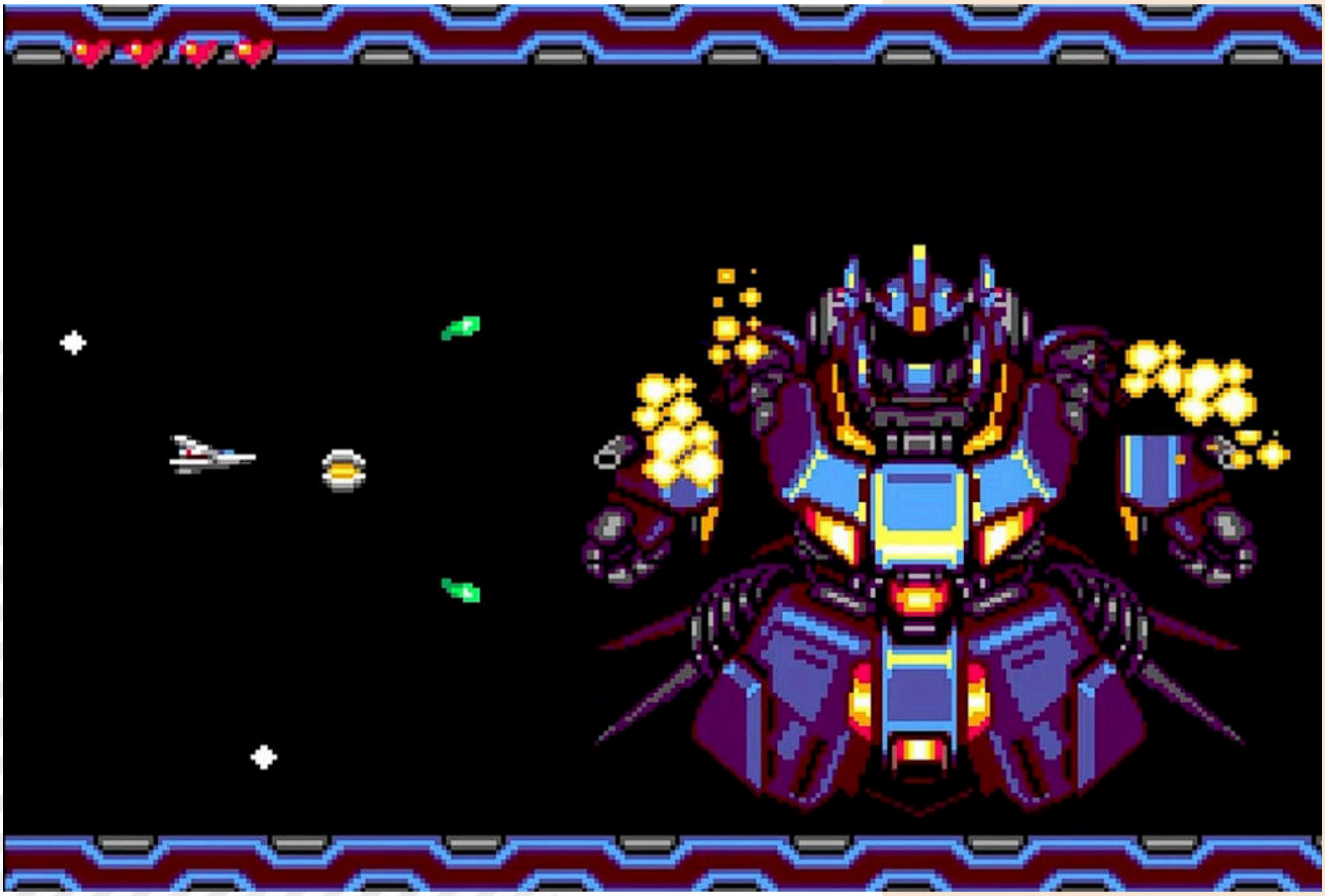
Lunar Skirmish is Earthion for Master System! A risky comparison, given that Earthion has not yet been released for Mega Drive (not physically), but we can do it.

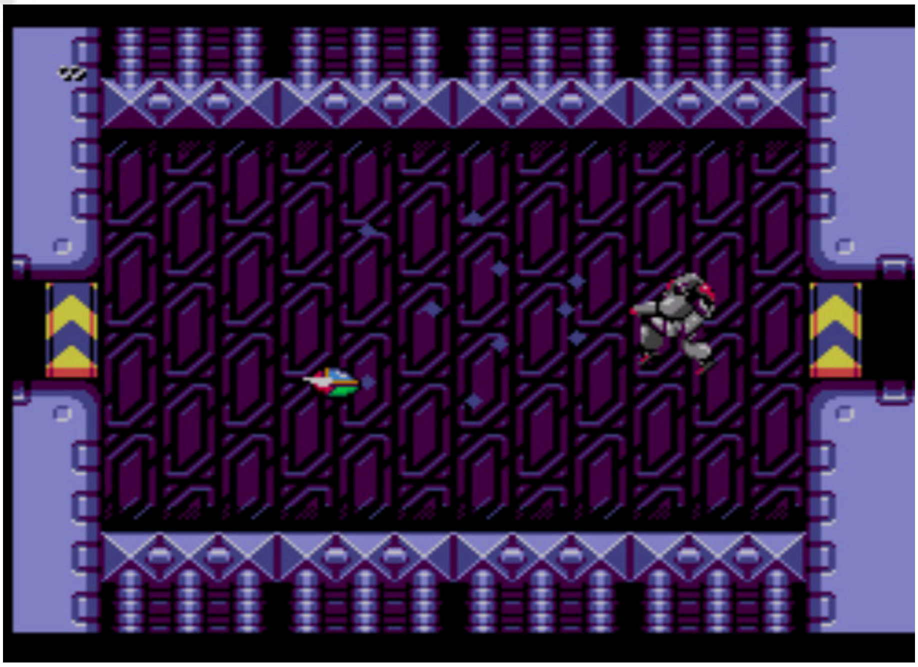
A shoot 'em up that has almost gone unnoticed but amazes with its technical quality and its "hard and pure" playability reminiscent of titles from the past.

The story is the same one we've seen for decades now: aliens attack a human colony in a distant galaxy, and we, the best pilots in outer space, have the arduous task of eliminating them.



There are seven well-detailed and distinctive levels, numerous enemies on screen, the much-feared boss





OUR FINAL SCORE

» Gameplay 95%

It has everything you need to have fun: power-ups, a great storyline, beautiful graphics, a gameplay system that alternates between classic style and exploration... and even the Konami Code.

» Longevity 90%

It's a difficult game for experienced players.

fights, and the final battle with a monstrous alien mothership.

Everything is reminiscent of and pays homage to the classics of the genre from our childhood.

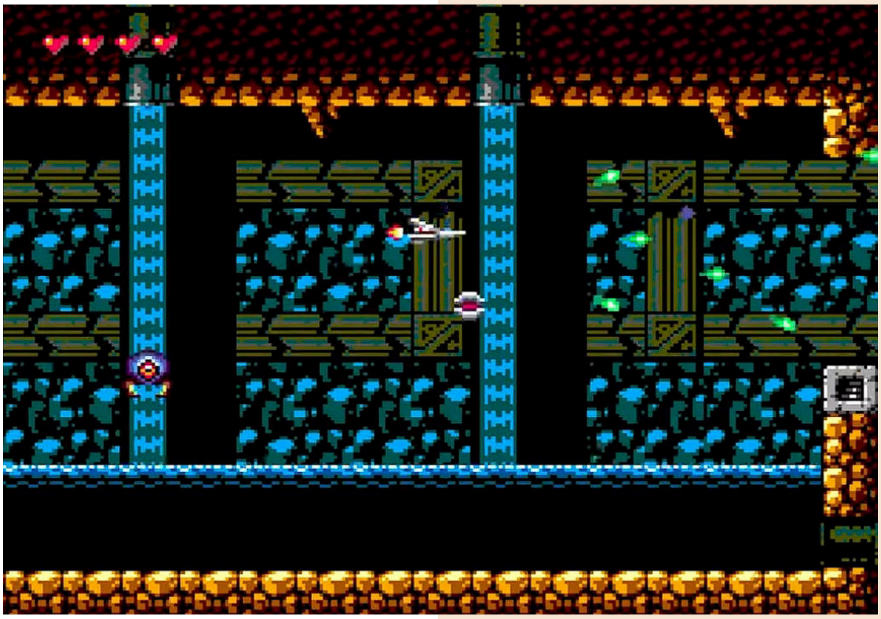
Lunar Skirmish uses some interesting technical tricks for smooth scrolling and parallax effects in different levels of the game. The developer has used some graphics libraries for scrolling and has done a really good job of customizing them to make the game very fast in both PAL and NTSC (the former runs at 50fps and the latter at 60fps). The work on the sprites is excellent, and the use of the SMS color palette is among the best for this console.

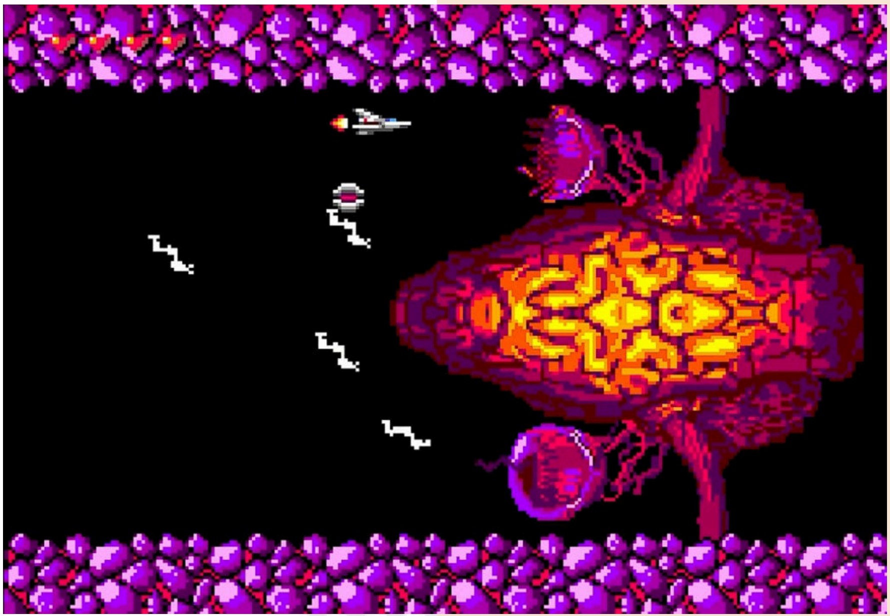
The game mode is also interesting, alternating between classic horizontal scrolling stages and operational sections of downward or upward exploration.

As in all shoot 'em ups, Lunar Skirmish also features the usual power-ups (speed is essential!) and even the inclusion of the Konami Code to unlock OPA OPA, the protagonist of Fantasy Zone, instead of the classic spaceship. This mix of traditional shootouts and exploration segments, combined with a high but never punitive difficulty level and first-class technical features, make Lunar Skirmish one of the best recent titles for Sega's 8-bit console.

A must-have game!

by **Marta Rossmann**



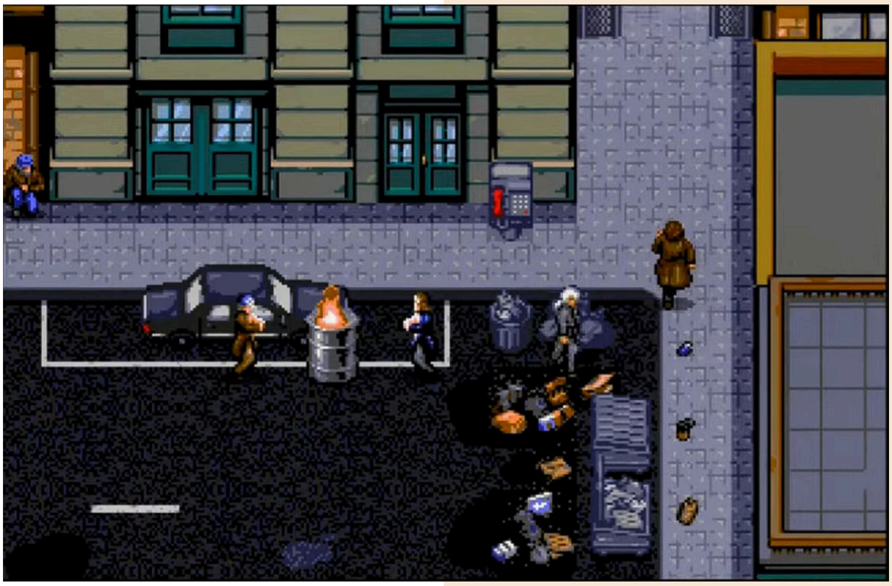


NEW GAME

THE SECRET OF THE FOUR WINDS

Year: 2025
Sviluppatore: Kai Magazine
Genre: ARPG
Platform: Sega Mega Drive
Website: https://kai-magazine-software.fwscart.com/SEGA_GENESIS_MEGADRIVE/cat4398490_555342.aspx

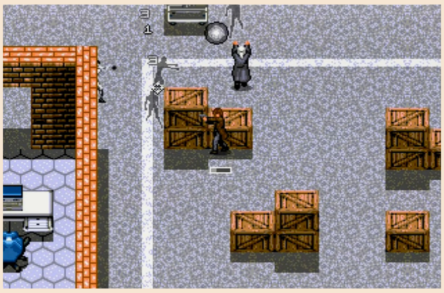
Kai Magazine has a single mission: to push the Mega Drive beyond its limits. I've been fortunate enough to follow all of their MD releases over the past five years or so, and each one has breathed new life into the beloved console, giving thousands of video game fans around the world one more reason to fire up SEGA's black 16-bit machine.



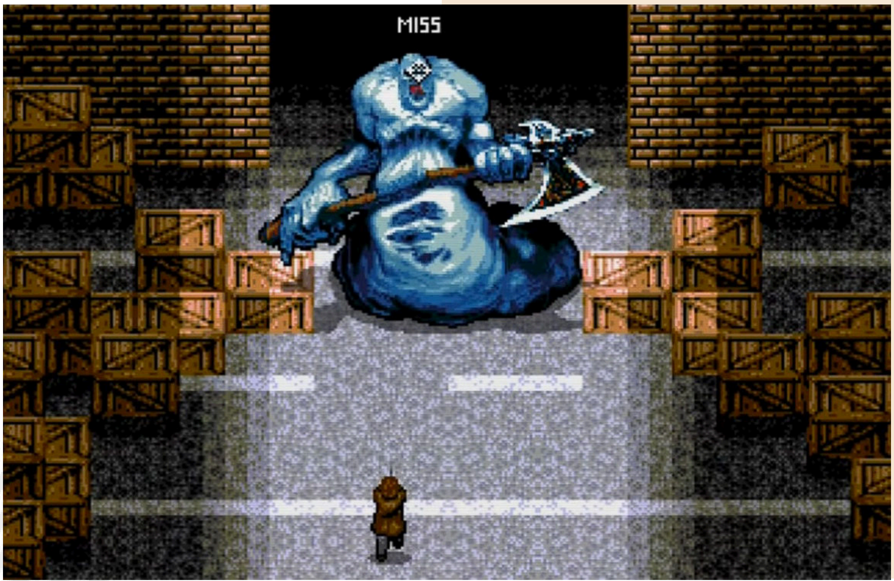
From the ironic MERCS-style Metal Dragon to the Metroidvania Life On Mars to the fantastic Life On Earth: Reimagined, the development house has pushed the hardware in different directions and crossed different genres, but none as "intensely" as The Secret of the Four Winds.

Starting with Inspector Henge chasing a suspect (oddly, he runs on foot while Henge chases him in a car!), the two end up in an abandoned warehouse, where the fugitive disappears and Henge watches a man die, steals his gun, and discovers that it is a supernatural weapon that kills ghosts. Bam! There couldn't be a better start for an MD title, helped immensely by the anime-style cinematic scenes (more on that later) and the incredible music.

Honestly, when the game started for the first time and I was able to choose the enhanced music over the standard one, I stood up listening to the Blade Runner-esque soundtrack, rich in sax and guitar, during the intro; a clarity and audio style I had never heard from the Mega Drive in thirty-five years of honorable service.



Apart from the soundtrack, the graphics are functional and depict a world with a top-down view in which the characters you control (a choice of three in total) move with a rather original movement system. One button is used to run (and consumes energy), while the others are used to aim and shoot. A bit like in some "modern"



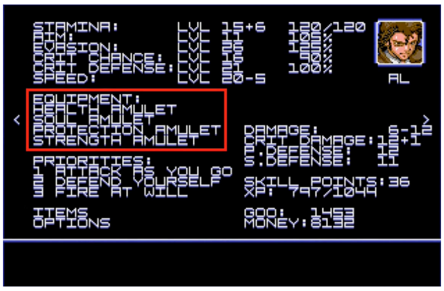
titles. Initially, it's disconcerting for a title on the 16-bit Sega, but as you delve deeper into the settings (mainly nocturnal), you get used to it and it works great.

While some levels require simple investigative activities and puzzle solving, the vast majority of The Secret of Four Winds consists of fighting your way through hordes of enemies, most of whom are weak in the light and much stronger in the dark.

What makes the game exciting is the way it blends action and role-playing, with character upgrades and skill increases. It's a challenging system, but definitely functional. The visuals are reminiscent of two titles in particular: Kojima's The Snatcher and Shadowrun, two impactful titles for all lovers of the genre.

Kai Magazine Software is one of the most exciting developers around for Mega Drive, and this title deserves a place on the top shelf of your game library. A game with retro Japanese aesthetics, amazing sound, and gameplay that may seem repetitive but will thrill all lovers of this genre. Buy it, the physical edition is killer!

by **Roberto Del Mar Pirazzini**



```
STAMINA:       LVL 15+6  120/120
AIM:           LVL 11    105%
EVASION:       LVL 26    125%
CRIT CHANCE:   LVL 16     90%
CRIT DEFENSE:  LVL 21    100%
SPEED:         LVL 20-5              AL
EQUIPMENT:
HEALTH AMULET
SOUL AMULET
PROTECTION AMULET      DAMAGE:       6-12
STRENGTH AMULET        CRIT DAMAGE:12+1
                       P:DEFENSE:    12
PRIORITIES:            S:DEFENSE:    11
1 ATTACK AS YOU GO
2 DEFEND YOURSELF      SKILL POINTS:36
3 FIRE AT WILL         XP: 797/1044

ITEMS                  GOO: 1453
OPTIONS                MONEY:8132
```

OUR FINAL SCORE

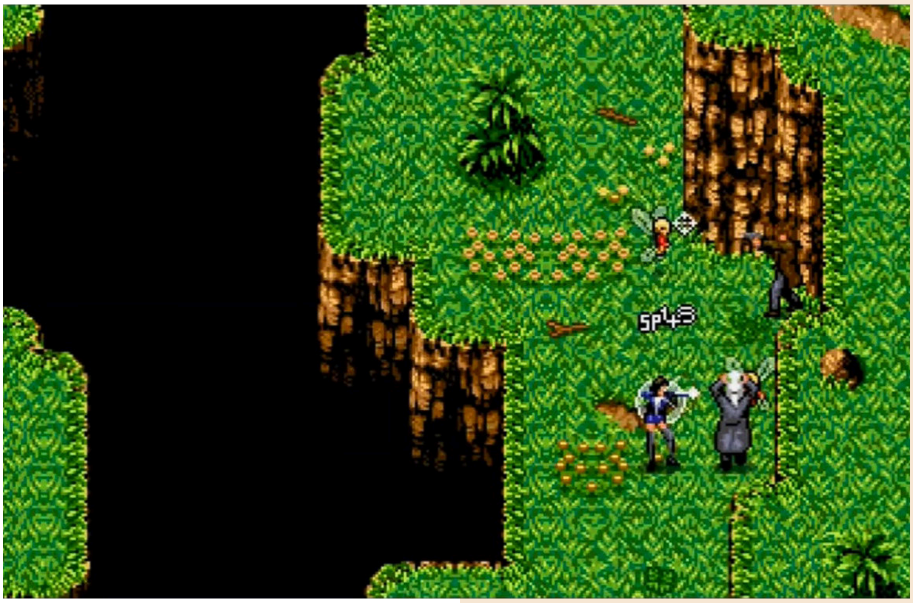
» Gameplay 90%

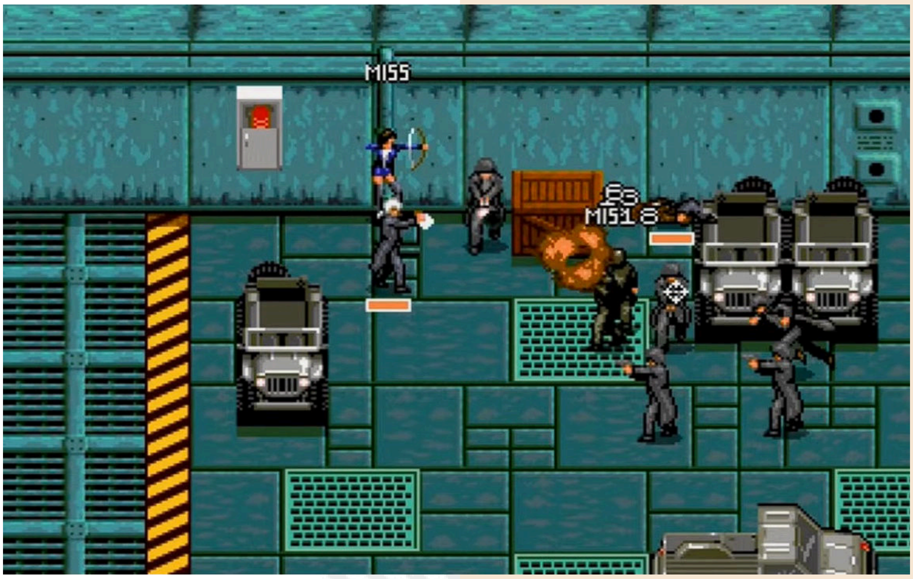
I like the "alternative" control system and the story dynamics that mix action with puzzles and classic role-playing.

» Longevity 90%

A must for fans of the genre. If you have a Mega Drive, save up your money and order it.

THIS IS INSPECTOR HENGE IN PURSUIT OF THE SUSPECT.





NEW GAME

KNIGHT ON THE MOON

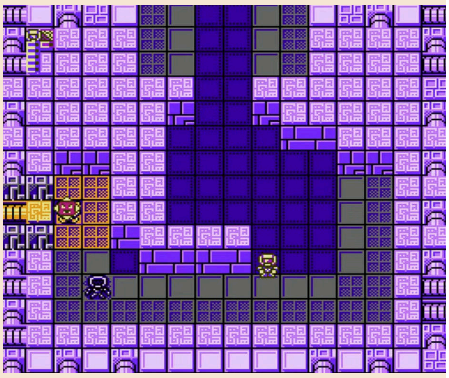
Year: 2024
Editor/Developer: Kn56k
Genre: ARPG
Platform: Nintendo NES
Website: https://kn56k.itch.io/knight-on-the-moon

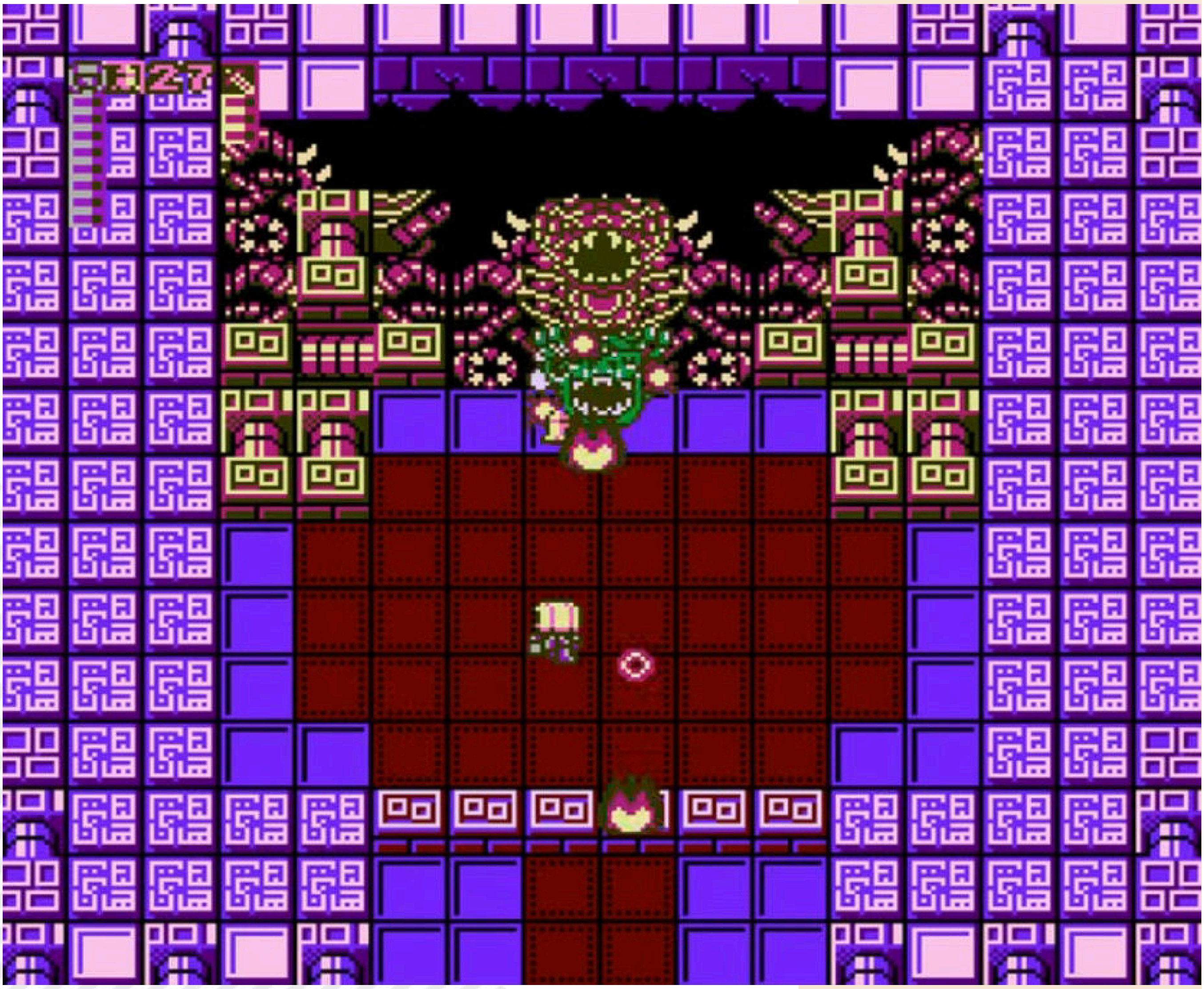
Guys, this game is awesome! A beautiful tribute to the golden age of NES video games. A title that mixes Zelda with Metroid and throws in a dash of Contra III. And it does all this with beautiful aesthetics and excellent level design.

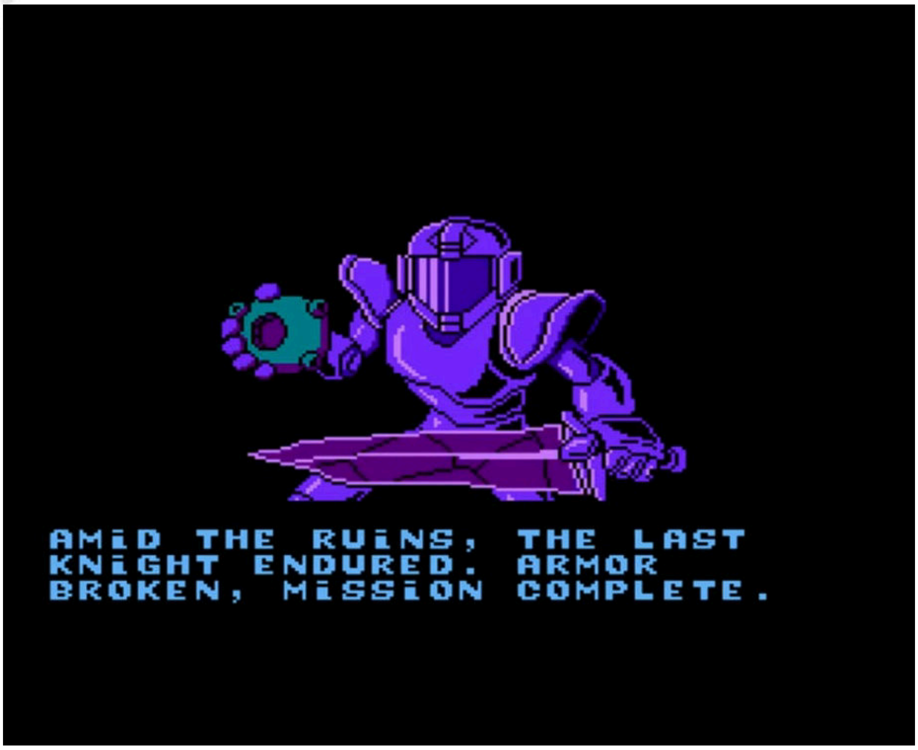
It starts with our protagonist (just like Samus, she's also a girl) making an emergency landing on an unknown planet. The view is from above and the style of play is very reminiscent of the first Zelda. Little is known about our mission. All we can do is try to escape and repair our vehicle, perhaps gathering information scattered throughout the wild world we find ourselves in.

Exploration and discovery are key. The only way to salvation is to find out where the weapons, secret entrances, and giant dungeons are hidden.

In this respect, it is very reminiscent of the aforementioned Zelda for NES, but there is also a lot of Metrodi.







AMID THE RUINS, THE LAST KNIGHT ENDURED. ARMOR BROKEN, MISSION COMPLETE.

Levers to operate, tunnels to explore, and equipment reminiscent of Nintendo's heroine Samus.

The worlds to explore are bizarre and very colorful. I must admit that I was impressed by the aesthetics and colors of this game. A beautiful color choice for the NES. The soundtrack is also worthy of praise.

The gameplay is quite fast-paced but simple and compelling. Nothing is left to chance, not even the simplest of boss fights.

Our main weapon is a lightsaber, but bombs and, above all, the rocket launcher will be essential.

It is worth noting some frame rate drops in certain areas that manifest themselves in slowdowns, nothing extreme but it is worth mentioning.

The real weak point is the sudden 'strange' spike in difficulty. Some levels and dungeons will be a walk in the park, others a living hell. I would have liked something more gradual, more similar to Link's adventures.

Apart from this minor issue, the game is one of the best modern ARPGs for the NES in terms of quality and gameplay and is well worth buying. It costs $10 but is well worth it!

by **Ingrid Poggiali**

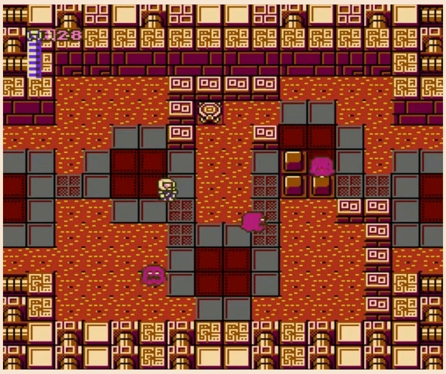
OUR FINAL SCORE

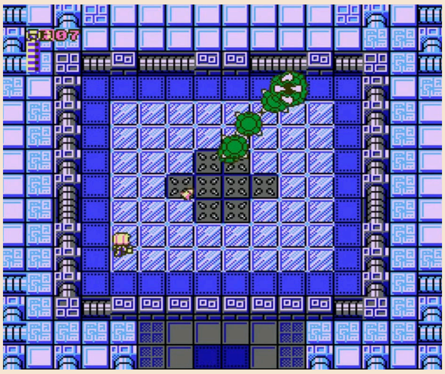
» Gameplay 90%

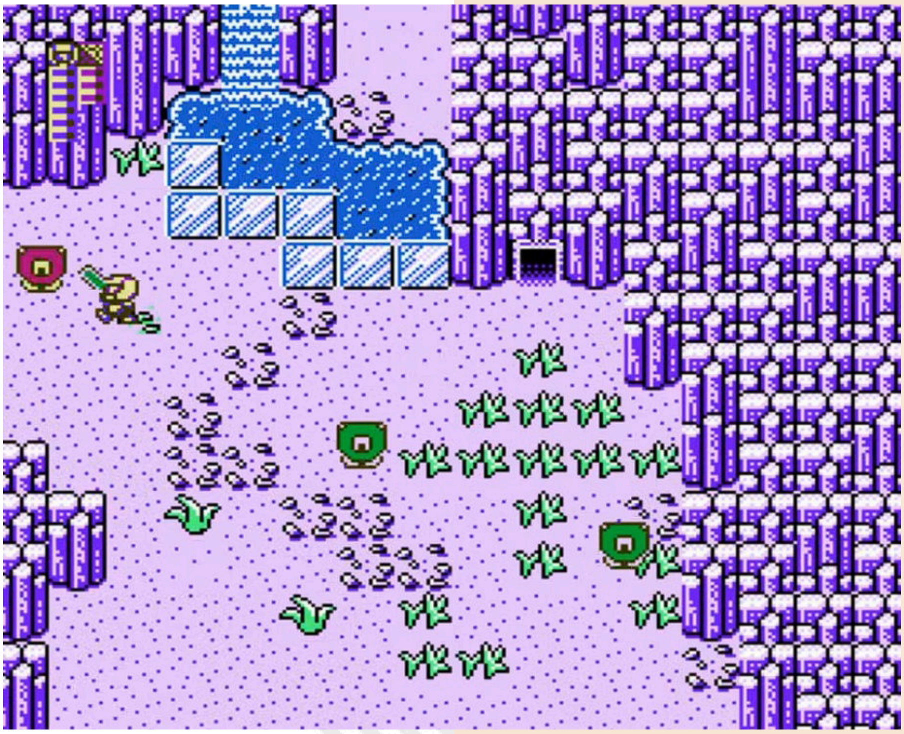
The game system is as simple and fun as the action on screen.

» Longevity 90%

The story flows well, but I would have liked a more "reasoned" difficulty level.







NEW GAME

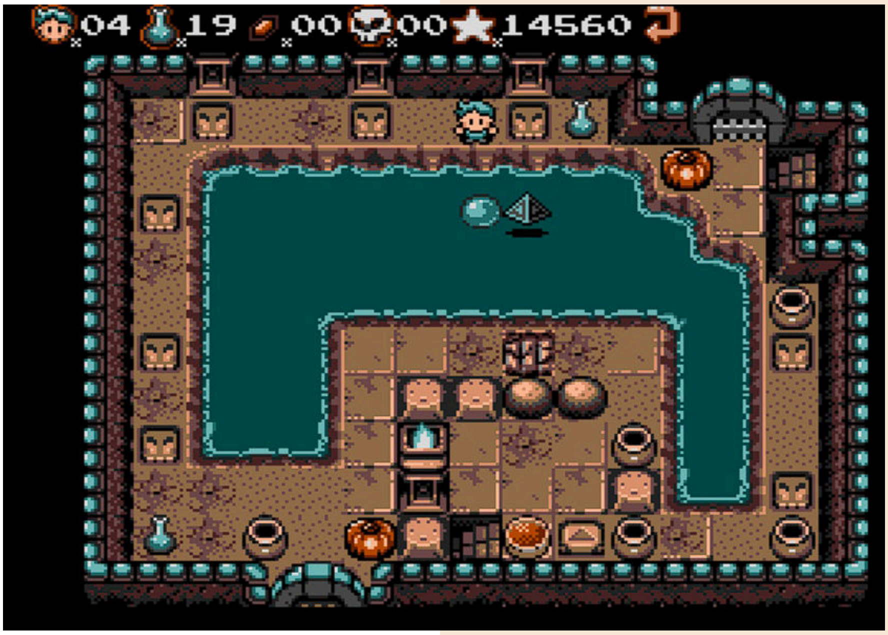
TINY MAGIC

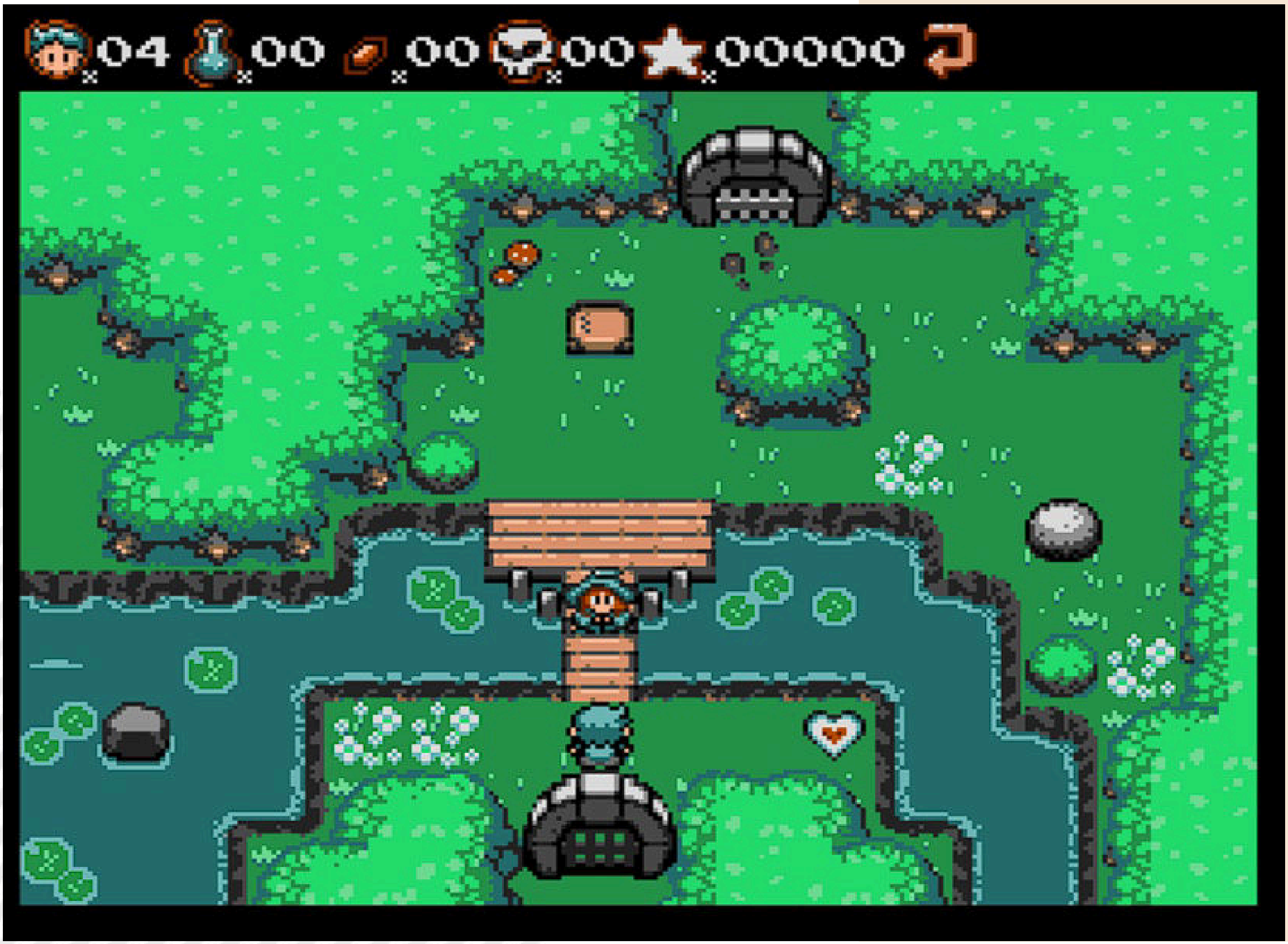
Year: 2023
Editor/Developer: Abyss
Genre: Puzzle/Avventura
Platform: MSX 2
Website: https://cotegamers.com/shop/en/accueil/176-tiny-magic-preorder.html

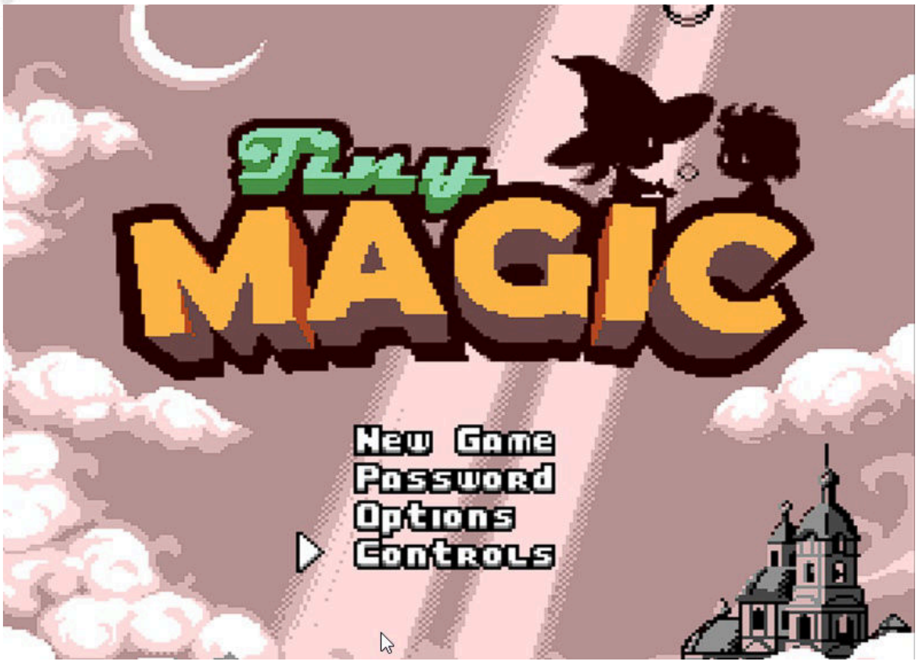
The year is 1890. Magic and wizards are disappearing from the world. At Mahoutokoro School of Magic, one student is chosen for the Master Wizard exam.

This year, the young water wizard Thiharis will be tested to assess his magical abilities and intellect. His teacher Alcina will guide our hero to the island of Ogaisuma for the final test. Will he be able to overcome the terrible trials and become the greatest wizard of his era?

This is the beginning of Tiny Magic, a puzzle game with adventure elements published by CotÃ© Gamers and developed by Abyss. A title released at the end of 2023 for MSX 2 that







OUR FINAL SCORE

» Gameplay 92%

Very appealing and dynamic. The puzzle/adventure mix works well and is enjoyable to play.

» Longevity 95%

Fifty levels is quite a lot, and the difficulty is well balanced.

amazed me with its quality and gameplay.

There are 50 levels with increasingly complex puzzles, and it uses a simple but compelling concept of combinations of rocks and switches to guide the protagonist through each level. It is reminiscent of Sokoban, but with very colorful graphics inspired by adventure classics such as Zelda.

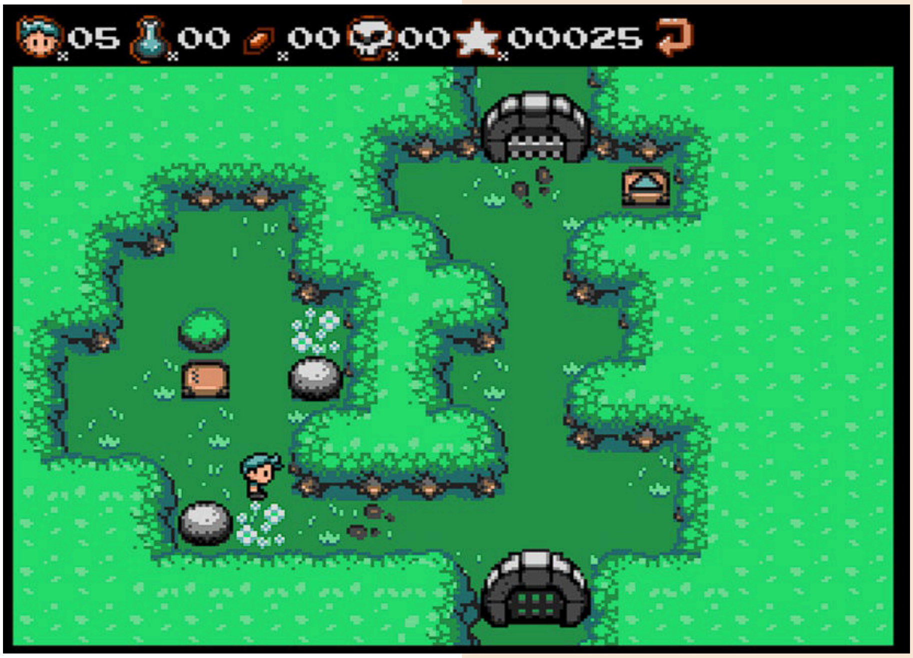
Players can also use magic bubbles and other resources to solve the puzzles in the level, and there is an error correction system. Not only are there rocks to move, levers, and switches, but also a good number of enemies to face and numerous traps. It's a very dynamic puzzle game compared to the usual.

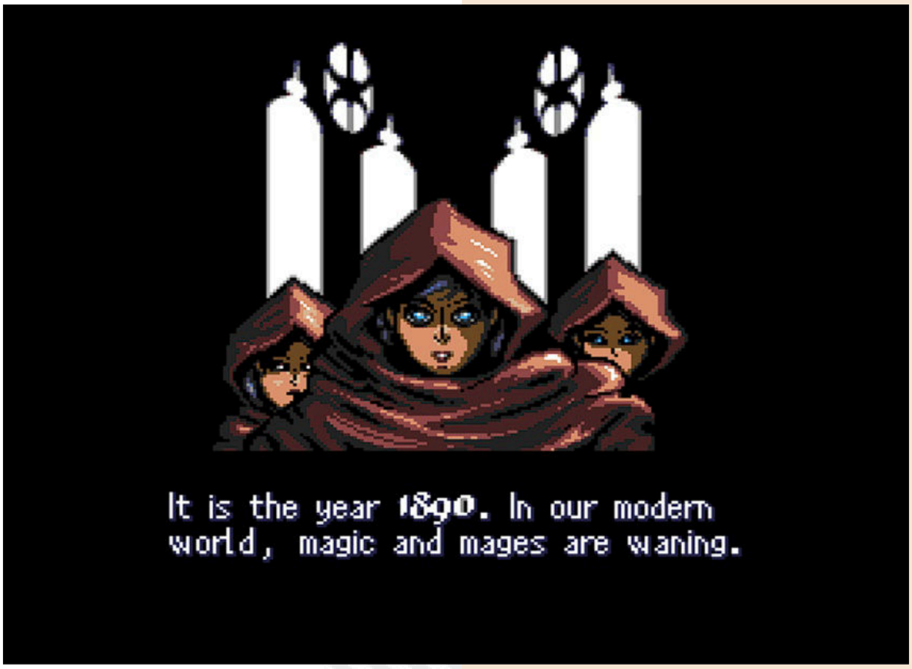
The game runs on MSX 2 with MSX-Musica, and it's really nice to look at and, above all, to listen to. The soundtracks are well done and atmospheric. The fluid and well-designed gameplay captivates from the start, thanks in part to a nice initial presentation and an excellent tutorial for learning the game controls. The password system allows you to resume the game from where you left off.

The game is available on cartridge with a hard case and a nice color manual. Great game!

Well done, Abyss.

by **Giampaolo Moraschi**




It is the year 1890. In our modern world, magic and mages are waning.

NEW GAME

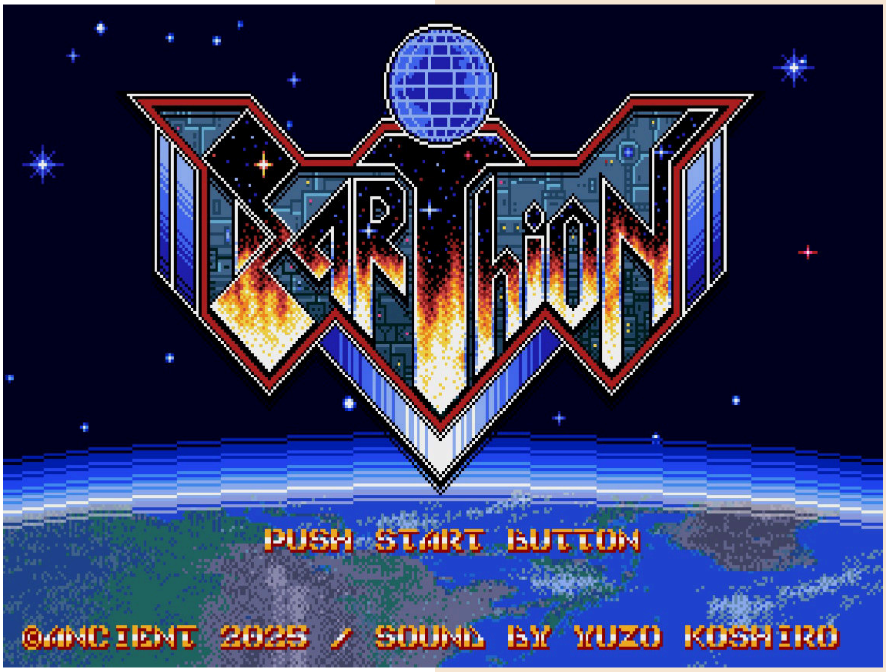
EARTHION

Year: 2025
Editor/Developer: Limited Run Games, Ancient
Genre: Shoot'em up
Platform: Steam – coming up in 2025 on Xbox, PS4/5 and Nintendo Switch and in 2026 on cartridge edition for Sega Mega Drive

Yuzo Koshiro needs no introduction. He is a legend in video game music. He took the "relatively" modest audio hardware of Sega's 16-bit console and made it do things no one thought possible; his soundtracks for titles such as Revenge of Shinobi and Streets of Rage are still considered classics today.

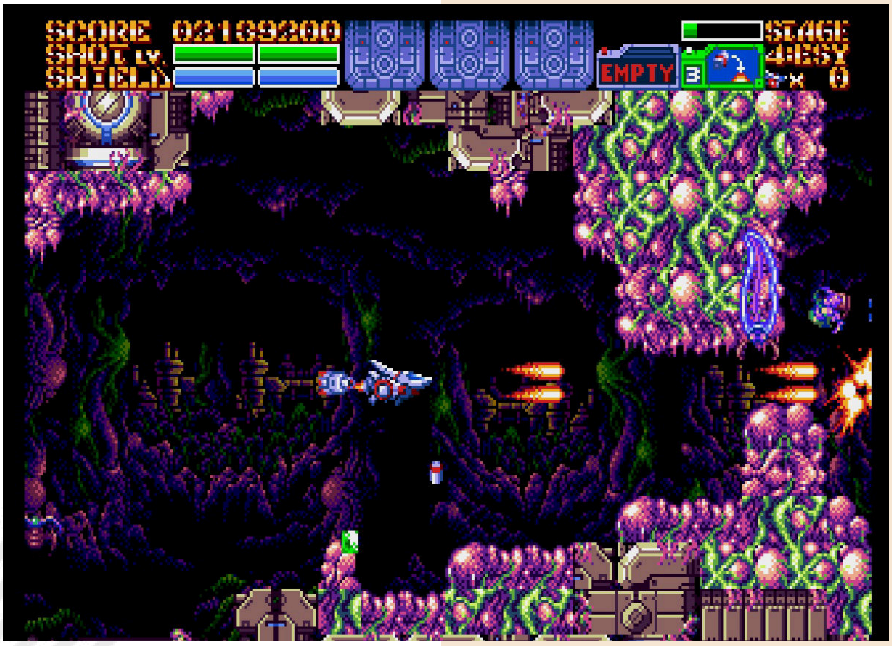
But his career has gone far beyond soundtracks for Sega: he has worked on Nintendo consoles with the soundtrack for Actraiser (an absolute masterpiece) and the Etrian Odyssey series, and we recently "heard" him at work with the soundtrack for Streets of Rage 4. He is also the founder (along with his mother and sister) of the development studio Ancient.

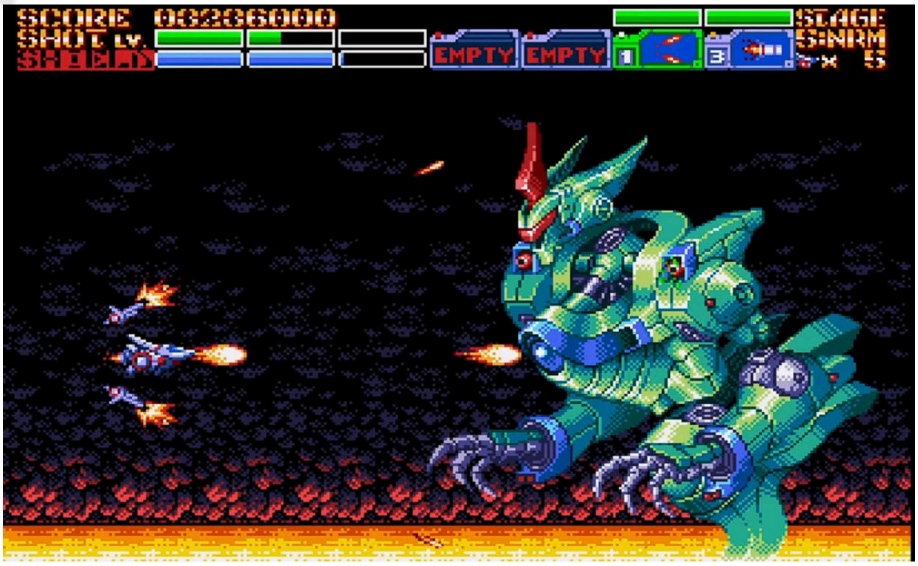
When this legendary man announced in 2023 that he was working on a new shooter called Earthion for his beloved Mega Drive, the anticipation among fans (myself included) was palpable. The demos, photos, and videos showed the unthinkable: from rotating graphics to vivid colors, from super-detailed animations to a killer soundtrack. Now the unthinkable has been released on Steam, and it absolutely lives up to its potential. We are probably looking at the best shooter available on Mega Drive.



Its physical release has been delayed until 2026, but all versions released to date (including the Steam version I am reviewing, which was launched first) run the game in emulation, so it's simple: what you see on a modern screen is the version we will see on cartridge (the game contains the ROM). Extremely... easy to play!

The main reason for this game's success is that it keeps things relatively



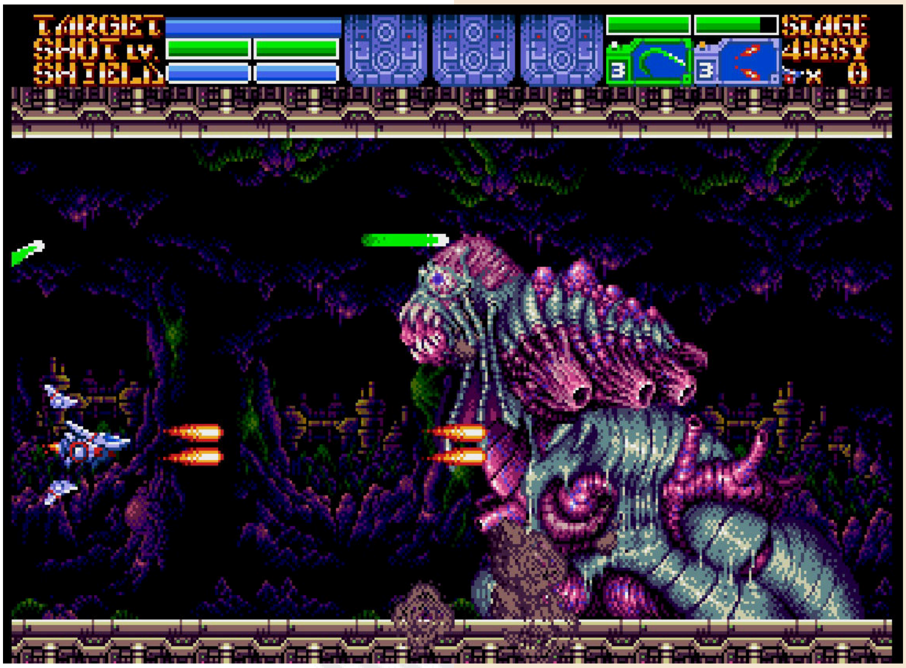


simple in terms of gameplay. The spaceship we fly has two main offensive options: the upgradeable standard shot and secondary weapons, which we can collect within the game levels. It is also possible to keep two secondary weapons and switch between them at will; these range from powerful lock-on laser missiles to eight-way shots to devastating depth mines (space mines). We can upgrade the types of weapons by collecting green gems called Solrium, dropped by dead enemies. The aircraft can withstand several hits thanks to its shield system, and if we manage to avoid being hit a second time, the shields will recharge. This is an interesting system which, when combined with the traditional life reserve, creates a rather forgiving experience... at least in the early stages.

It is interesting how Koshiro and Makoto Wada (director, designer, and developer of the game) managed to focus their attention on creating a truly sublime shooter.

The eight levels of the game are varied and captivating, even when replayed over and over again. the initial phase takes place during a massive space battle between rival fleets. We therefore find ourselves shooting down hordes of enemies and avoiding debris from exploding space cruisers. Each area features the classic mid-level bosses and one (of varying sizes) at the end of the level. Some are more visually impressive than others, but they all represent an acceptable level of challenge.

We then have organic, terrestrial (the level with the city covered in vegetation is beautiful) and space levels. All presented in an original and innovative way, testimony to Wada-San's extraordinary artistic level, refined with titles such as Protectors: Cart of Darkness and Royal Anapoko Academu. To say that this is one of the most beautiful Mega Drive games would be an



understatement; it is a visual delight in every respect.

It is also one of the games with the best sound, but then again, I never doubted that. Yuzo Koshiro is the undisputed master of Mega Drive audio hardware and, with this game, he has composed what could be considered his magnum opus on 16-bit, even if the tone is slightly different from the techno-style soundtracks seen in the Streets of Rage series.

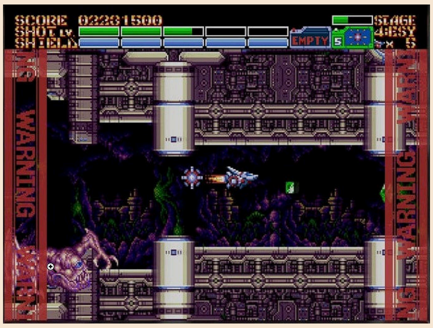
There is a sort of inspiration/homage to classic shooter series of yesteryear such as Gradius, Darius, and Thunder Force, with lively and stimulating songs that allude to intergalactic evil and the epic nature of our journey to build peace. Some melodies are so "powerful" in structure that they seem to come from more powerful hardware, yet each song retains the classic FM sound distinctive to the Sega console, right down to the iconic drum beats and incredible vocal samples.

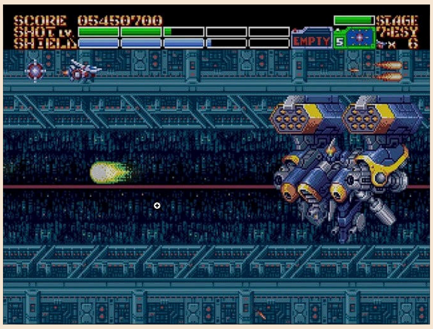
Now, as we all know, this type of game lives or dies based on its difficulty: if you make it too easy to finish in a single session (a problem common to several Mega Drive titles from the 1990s), the title quickly falls into oblivion. However, if you go in the opposite direction and make it too challenging, you risk alienating and destroying a wider audience.

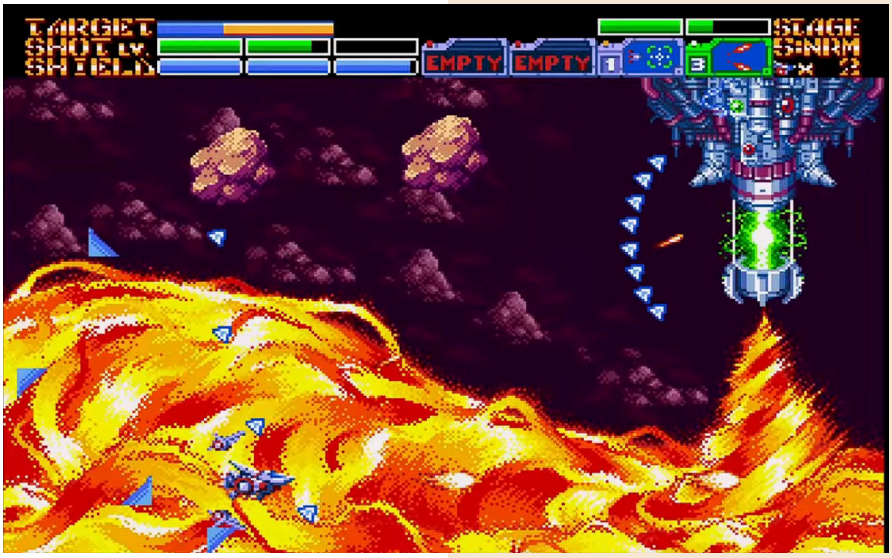
The solution in Earthion solves all this; very similar to the legendary Radiant Silvergun. The game allows you to keep your progress between sessions, transferring lives, power levels, and expansion slots via an old password system (I would have preferred a save system, but I'll let it slide this time).

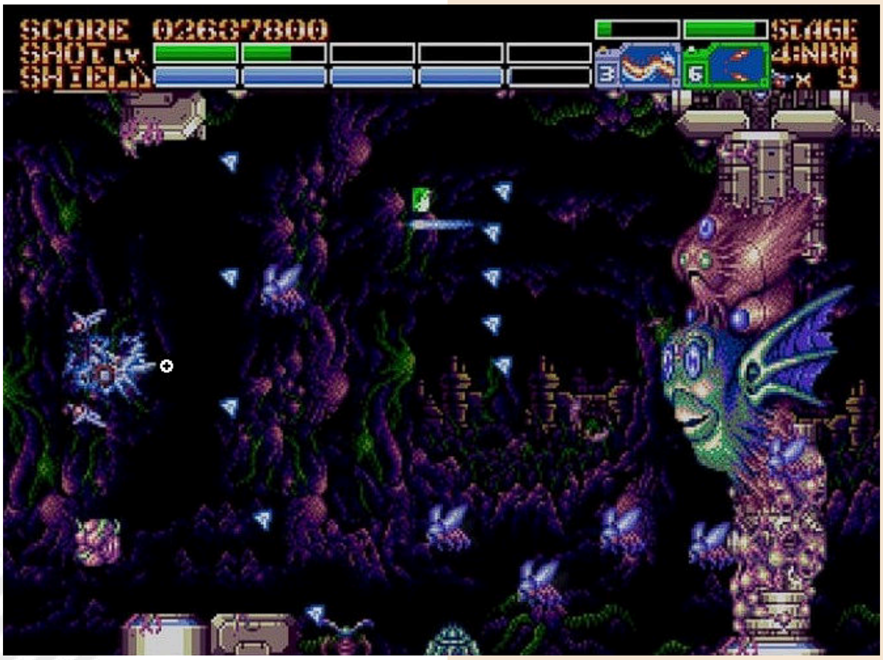
At this point, it's worth explaining how this system works. If we finish a level with a specific secondary weapon, we will have the opportunity to permanently upgrade an aspect of our ship.

We can increase the maximum power level of weapons (up to 10), add a new power/shield slot, or change the balance of slots so that we can keep more secondary weapons, but at the expense of a power/shield slot. We can choose to start the next stage with a new secondary weapon or add an extra life to our supply. This system allows newcomers to take on the game's challenge at their own pace, starting in "easy" mode and using Adaptation Pods to increase their ship's power before tackling the default "Normal" difficulty level. Once you become familiar









with this system (and I assure you that completing the game at this level is no small feat), you can switch to "Hard" or "Hoshot" mode, which significantly increase the challenge.

What makes this approach so appealing is that it is completely optional; if you are a veteran of the genre, you can ignore the upgrade path offered by the Adaptation Pods, just as you can choose to ignore the option to continue using credits when you run out of lives. Whichever direction you take the game, the longevity is remarkable, more so than your typical 16-bit shooter, plus there are special "Challenge" levels that offer unique level configurations and expect you to get the best score possible.

However, Earthion does eventually come to an end; even inexperienced players will eventually reach the conclusion in the short "easy" mode, so enjoyment will be determined by how much you enjoy immersing yourself in the game levels. Online leaderboards are also available on Steam, which helps to extend the game's lifespan (the latter will not be available in the physical edition on Mega Drive).

Earthion is everything I hoped it would be when Yuzo Koshiro first announced it: a challenging shooter that pushes Sega's 16-bit hardware to its limits in terms of graphics and audio, while combining a high level of polish with tight gameplay and plenty of longevity. On a system that has more than its fair share of quality shooters, Earthion stands out as one of the best, and while purists will have to wait a little longer to play the "real" version of the game on Mega Drive/Genesis, this emulated edition is perhaps the ideal way to try it out thanks to comforts such as screen filters and granular control over audio and video settings, as well as the addition of elements such as online leaderboards and game prototypes.

Some may complain about how outdated Earthion is compared to modern shooters, but these complaints miss the point: Koshiro and his collaborator Makoto Wada sought to create the best shooter possible for the Genesis/Mega Drive, constrained by the technical limitations of hardware that is now over 30 years old. They achieved this goal spectacularly, making it a must-have for fans of the genre and those who appreciate retrogaming in general.

by **Carlo Nithaiah Del Mar Pirazzini**

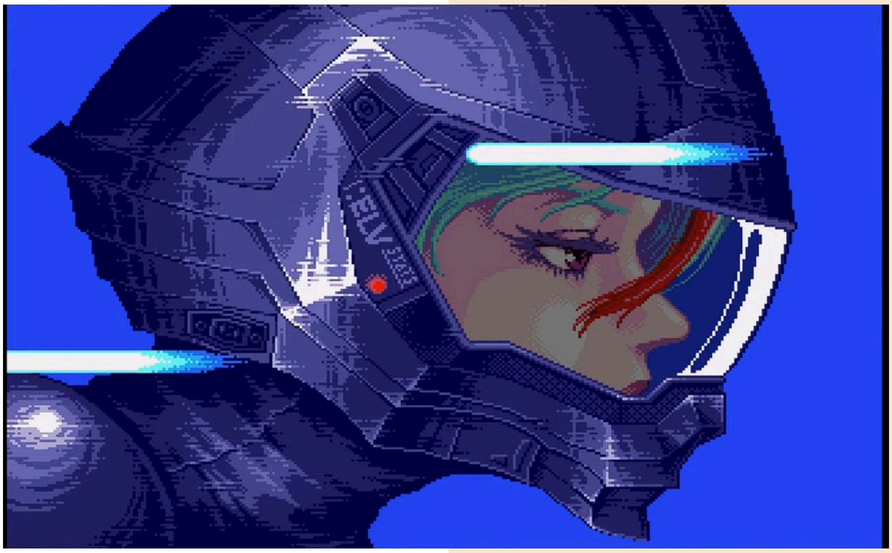
OUR FINAL SCORE

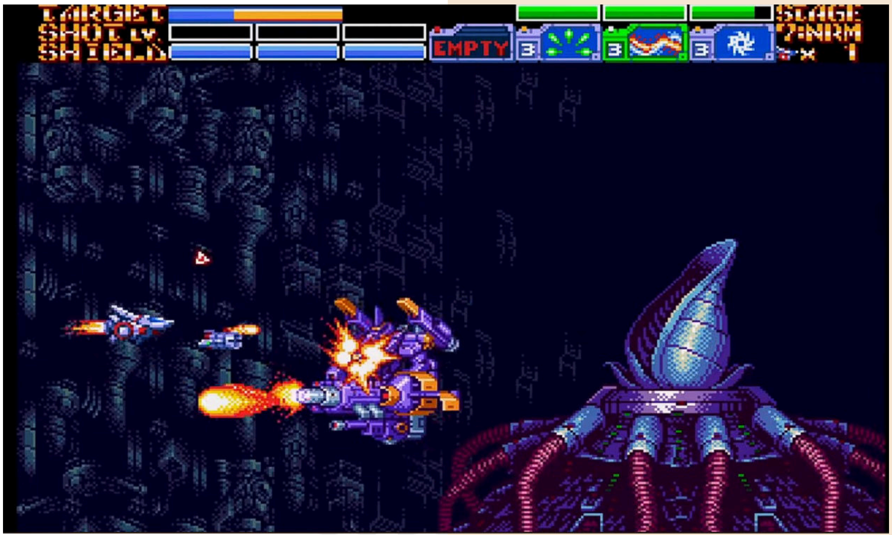
» Gameplay 95%

Technically stunning and featuring one of Koshiro's most beautiful soundtracks. Fast-paced, challenging, addictive gameplay and awesome level design!

» Longevity 95%

The password system is too "old-fashioned," but everything else is perfect, including the progressive system that makes the game accessible to everyone.





NEW GAME

CUBIX

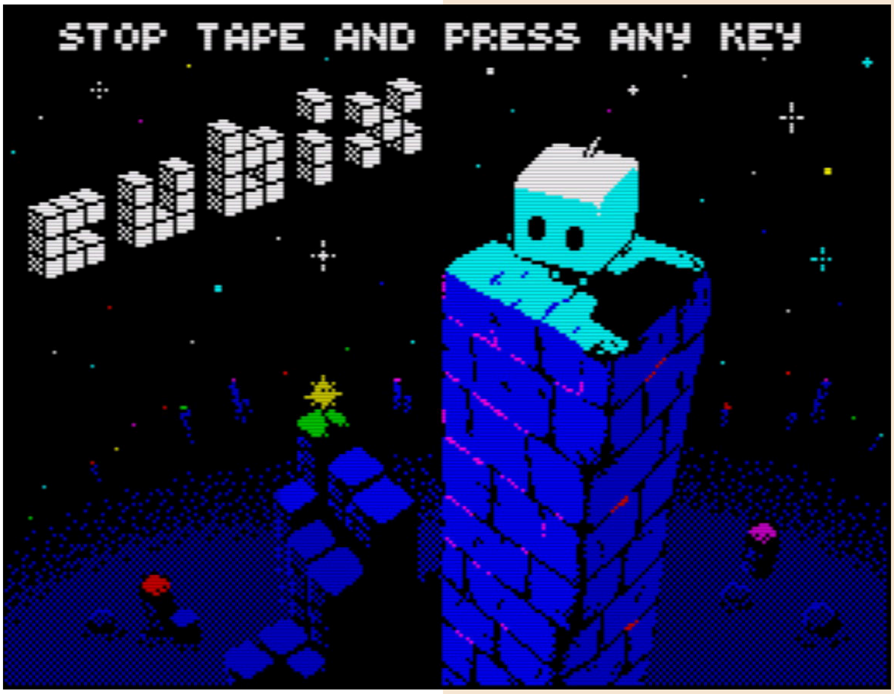
Year: 2025

Editor/Developer: Gogin

Genre: Platform/puzzle

Platform: ZX 128, +2,+2A, +2B o +3

Website: https://yrgb.ru/



In a distant parallel universe, geometry reigns supreme. This place is called Cubix, where the laws of physics are governed by symmetry and the world is made up of perfect cubes.

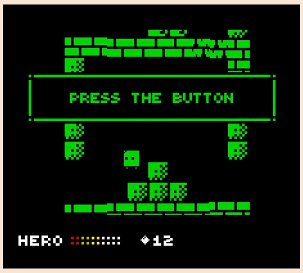
Marvelous towers rise toward the starry sky, cities shine in their bright facets. All this thanks to a precious Sacred Talisman, a mega magic die that radiates balance, order, and perfection.

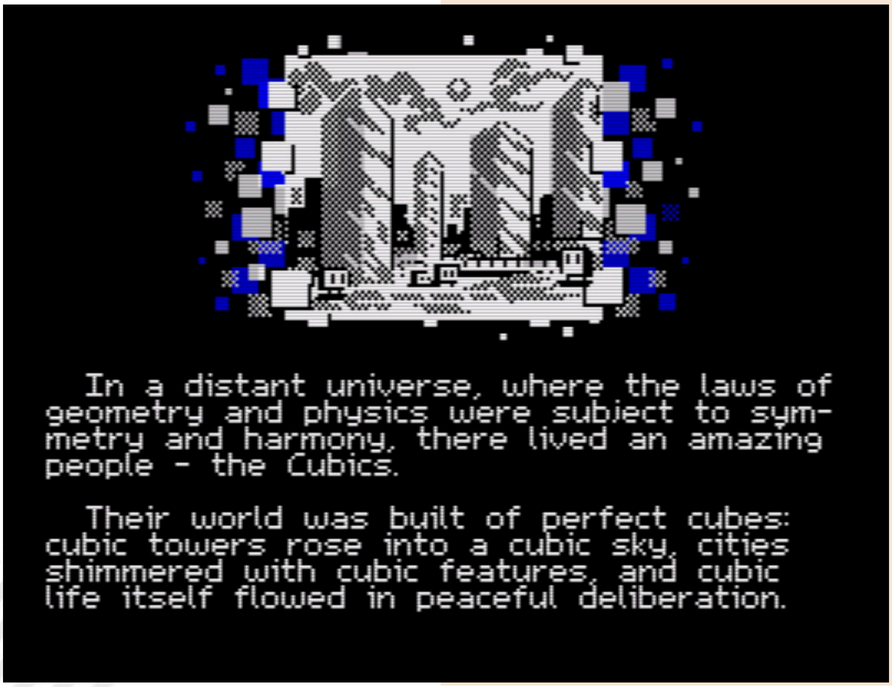
One day Hexatron, lord of disorder, appeared from the parallel Dark Dimension. A grim character who hated the perfection of cubes and wanted to eliminate harmony. He stole the sacred talisman, divided its faces and hid them at the top of the city's tallest towers, filling them with traps, mazes and... monsters.

And this is where we come in, playing as BIX, a simple cubic citizen who escaped Hexatron's clutches and whose task is to free the world of Cubix. Obviously, he will have to climb all the towers, overcome the treacherous trials, collect the faces of the Talisman, and repair it to restore Perfect Peace. That's all. Now I'm signing off and going back to playing this little masterpiece, I have no time to waste!

Joking aside, Cubix is a platform/ puzzle game with small elements of Metroidvania. A lethal combination with 6 large (really large!) game environments (the towers), cute but annoying enemies, and various logic-based puzzles. As the game progresses, little BIX will acquire new skills that will allow him to overcome certain obstacles more easily.

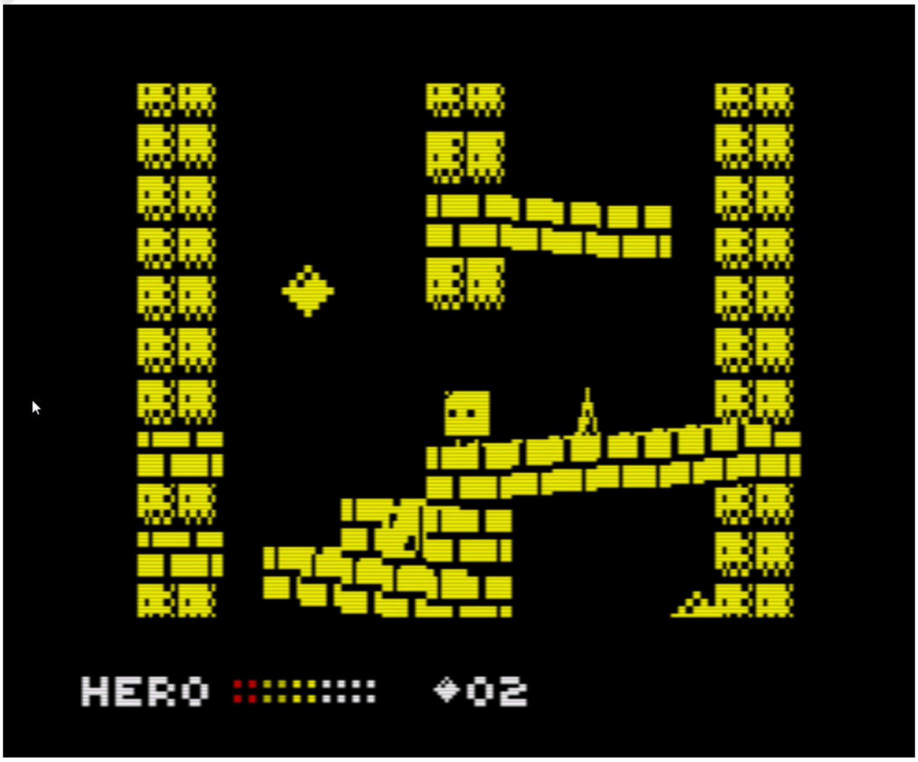
It is the first 3D platform game for the ZX Spectrum, and graphically it is incredible. The 3D engine used by





In a distant universe, where the laws of geometry and physics were subject to symmetry and harmony, there lived an amazing people - the Cubics.

Their world was built of perfect cubes: cubic towers rose into a cubic sky, cities shimmered with cubic features, and cubic life itself flowed in peaceful deliberation.



the game with true perspective is maniacal in its precision, speed, and perfection and was written entirely from scratch by the developer. The graphics are simple but captivating and animated with extreme care. It is undoubtedly a great work of technical and artistic precision. The monochrome is poetry in this game, from the beautiful graphic presentation to the ending. All this is accompanied by eight impactful and well-written tracks.

The game does not run on ZX 48k but requires at least a ZX 128, +2, +2A, +2B or +3, 3.5 MHz speed and 128 KB of RAM, to which we add the AY/YM chip for audio. Obviously, in emulation and on The Spectrum, it ran smoothly without too many problems.

The gameplay, although it takes some getting used to and understanding, is the strong point. The small Metroidvania elements add depth to a game that offers a great challenge. It can't be tackled without a little practice, but once you've learned it, you'll find it hard to put the joystick down during the challenge. To make everything more "modern," a "rewind" option has also been added to replay the level or the entire world (essential).

Perhaps it's not an easy title or one for everyone, but if we're still playing on a Spectrum in 2025, we're not exactly "rookies" just starting out, are we? Highly recommended.

A title that is in the running to win the Russian YRGB competition (link: https://yrgb.ru/ sign up and vote).

by **Carlo Nithaiah Del Mar Pirazzini**

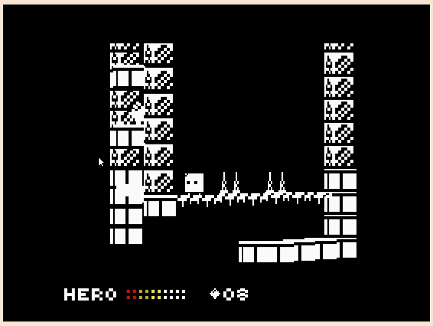
OUR FINAL SCORE

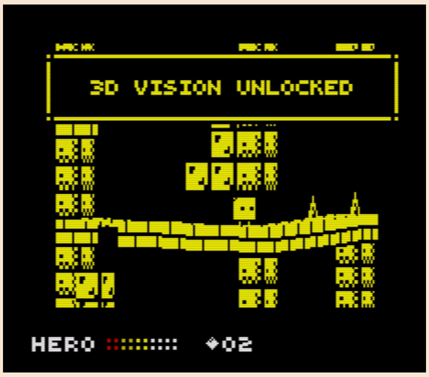
» Gameplay 95%

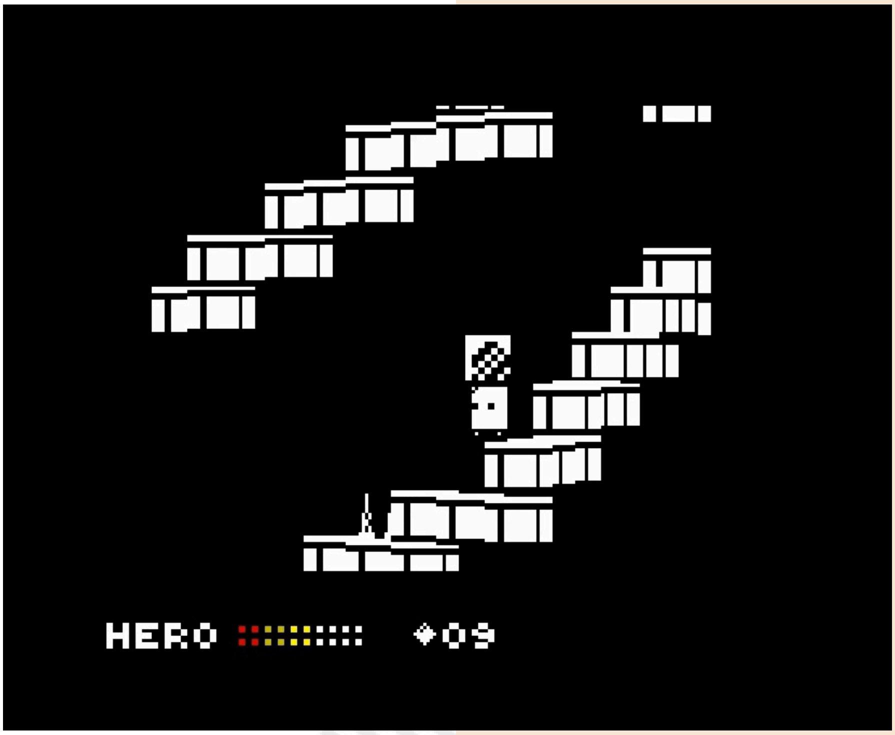
Everything in the right place! Intriguing mechanics, well-designed puzzles, and awesome technical aspects.

» Longevity 95%

Challenging, just the way we old-school hardcore gamers like it.







NEW GAME

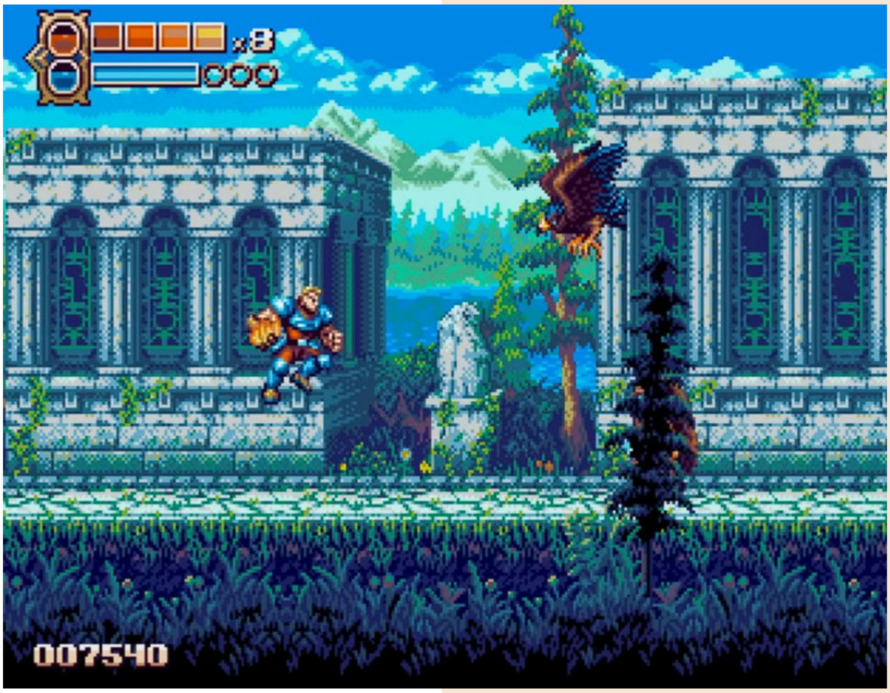
DAEMONCLAW: ORIGINS OF NNAR

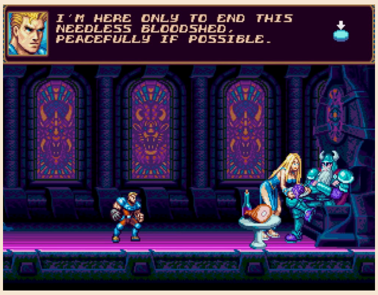
Year: 2025
Editor/Developer: Neofid Studios, BitBeam Cannon
Genre: Beat’em up
Platform: Sega Mega Drive
Website: https://neofid-studios.com/products/pre-order-daemonclaw-new-boxed-cartridge-with-printed-manual-for-sega-megadrive-genesis-i-ii-and-iii

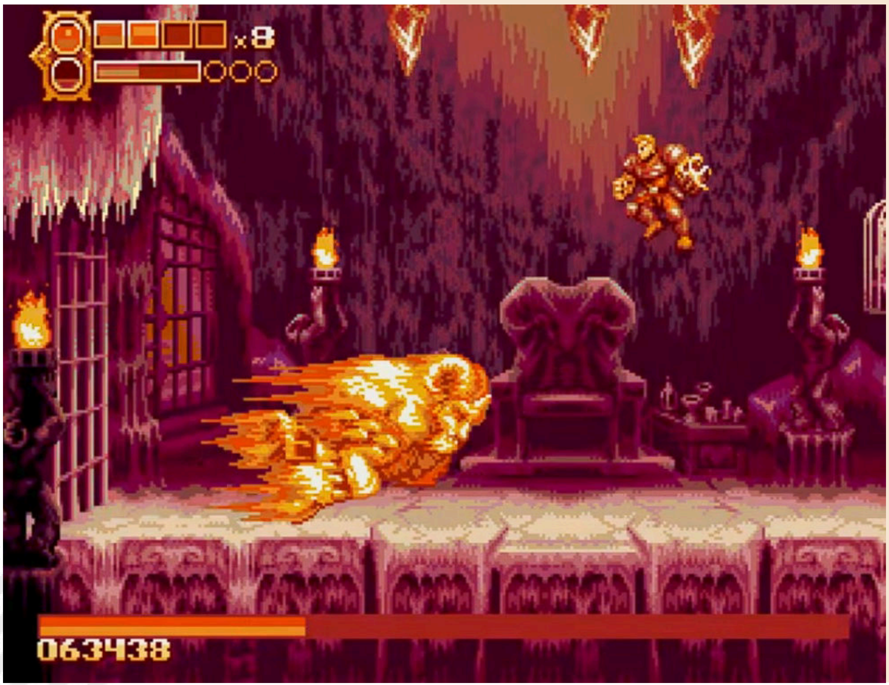
In this issue, you will find a whole host of new games for the Sega Mega Drive. Never before has the 16-bit Sega received so much new content. Some titles are absolute masterpieces, others less so. Either way, the Mega Drive scene is alive and kicking, which is great news for us.

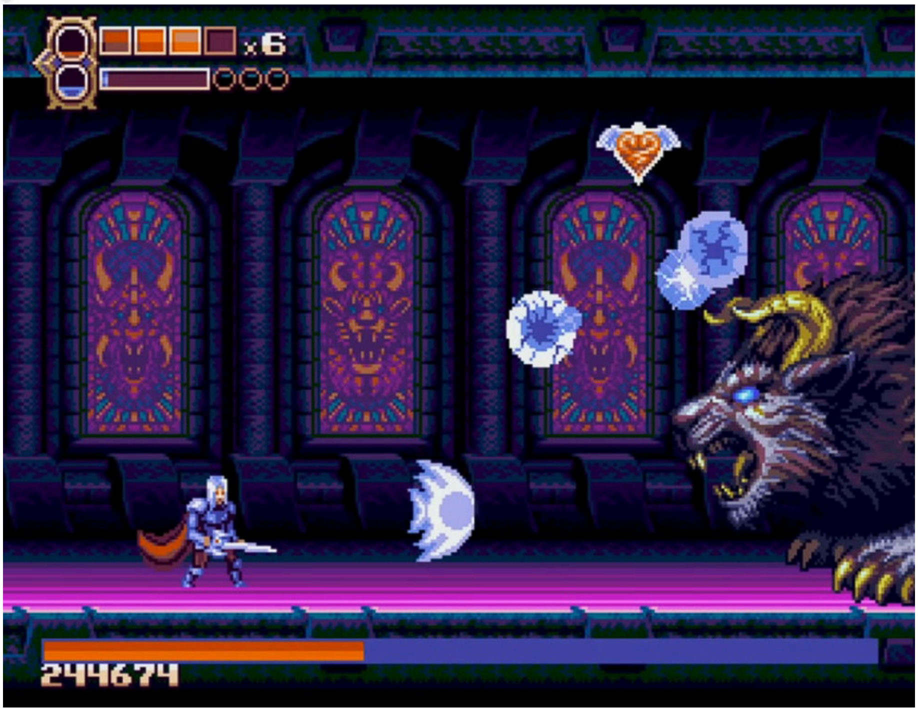
DaemonClaw: Origins of Nnar is a title we have been following for some time. Initially in development on Commodore Amiga (where it could run without any problems), it changed platforms and was released in three versions: one for STEAM, one for Neo Geo, and one for Mega Drive (the test version). In the game, we take on the roles of two characters. The first is the protagonist named Elrik Eisenfaust (a sober name). He is the hero with the arduous task of defending the lands of Nnar from the hordes of evil Zverkins... with his fists. Not just simple punches, but powerful, fiery magical blows, thanks to his enchanted glove. Eisenfaust is a typical big man from side-scrolling fighting games: physically fit, agile, and determined. Against him (even in easy mode), we will face hordes upon hordes of enemies from all directions, many of whom are equipped with shields and armor that must first be removed in order to inflict adequate damage. Our hero can parry blows and throw them back, use fiery uppercuts and aerial counterattacks, or neutralize enemies with a powerful diving punch. There are plenty of opportunities to 'bash' with extreme satisfaction, with something for everyone, but mastering them is a more complicated task. Fortunately, there is a nice tutorial that allows you to familiarize yourself with the game controls.

The other character (only present in the Mega Drive version, apparently) is Gareth, a hero already seen in




I'M HERE ONLY TO END THIS NEEDLESS BLOODSHED, PEACEFULLY IF POSSIBLE.





Demons of Asteborg. He is decidedly less 'physically fit' but equipped with a broadsword and different fighting techniques. This dual approach to the adventure gives the product a bit of longevity. This is an advantage, given that it is not a particularly 'long' title. Add to that an epic fantasy-style soundtrack composed by the legendary Allister Brimble, and the package already looks impressive.

Aesthetically, it has that typically "European" flavor of home computer productions. The graphics are pleasant, and the colors are beautiful, vivid, and rich. However, there is a lack of variety in the enemies (they are all the same), and we noticed some imperfect animations in certain characters. The various bosses, on the other hand, are interesting. They are well characterized both aesthetically and in their approach to combat.

So, you may be wondering, why are the ratings so... low? Because it has two fatal flaws for a cartridge title costing â,¬70. The first flaw is definitely its longevity.

The two-character option is nice, but the game has very few levels (5 levels), too few. The differences between the characters in the game are only stylistic; playing the game with Elrik or Gareth doesn't change much in terms of the levels.

The second flaw is in the gameplay. DaemonClaw is not a refined product. The gameplay is always the same (except for boss fights, of course). The enemies always behave in the same way: monsters on foot with weapons, monsters on foot with ranged weapons, aerial attacks. This is repeated in all levels. Boring.

Finally, let's add a certain "woodenness" to the protagonist's control system. Nothing terrible, but in some situations it is frustrating not to be able to move with "agility" to get out of a sticky situation.

It's a real shame, because on paper the title looked promising. Years and years of development for something that is short-lived and boring. What a pity.

We don't feel like rejecting it, but it's certainly not the title we've been waiting for all this time, nor the one that was promised. Expectations are a nasty beast.

by **Carlo Nithaiah Del Mar Pirazzini**

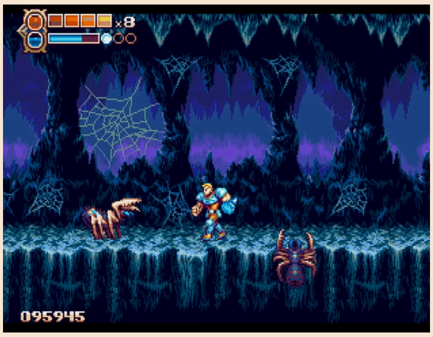
OUR FINAL SCORE

» Gameplay 70%

The two characters are interesting, but the game is "wooden" and with repetitive patterns in all levels that take away the fun and make it boring.

» Longevity 60%

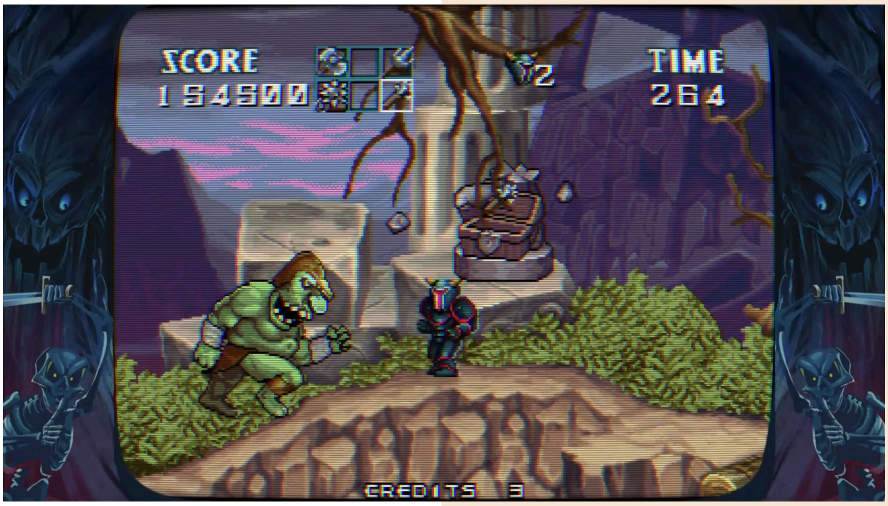
The game can be completed (after a little practice) in just over an hour... playing first with one character and then with the other... which means 30 minutes of play per character. Not much, very little.



NEW GAME

GLADMORT

Year:2025
Editor/Developer: PixelHeart, Chiponsteroids
Genre: Platform
Platform: Neo Geo, Dreamcast, STEAM, Xbox, PS4/PS5, Nintendo Switch
Website: https://www.pixelheart.eu/en/home/

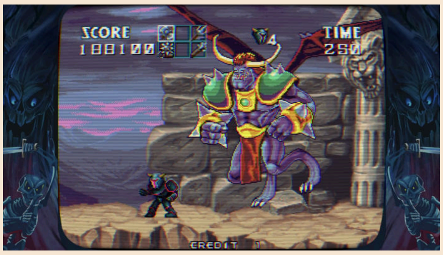


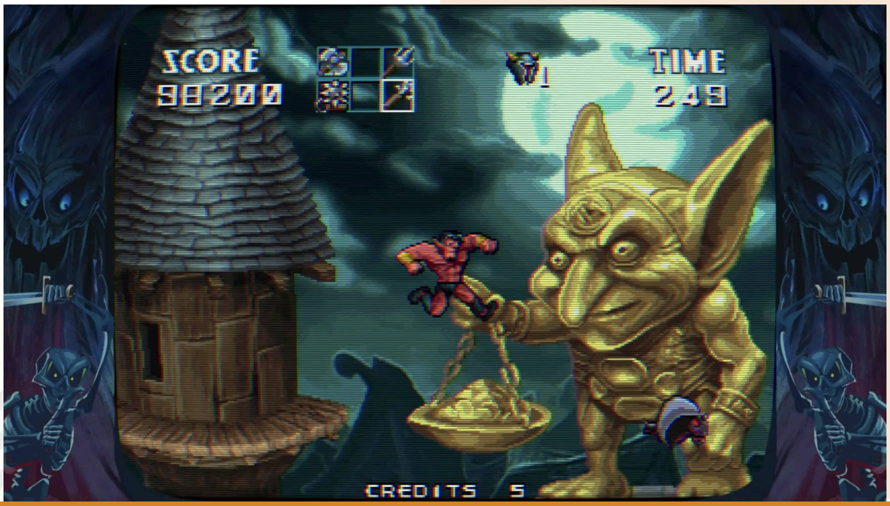
GladMort is a new game designed and published for NEOGEO, Dreamcast, and new platforms in 2025. The publisher has actually made physical cartridges available for purchase, which is really cool, but like most NEOGEO games, the physical version of GladMort is WAY more expensive than a typical game in 2025 (even with recent announcements of $80 games), retailing for â,¬379.90. Fortunately, it's also available for $12.99 on PC/Steam and $14.99 on PlayStation, Xbox, and Nintendo Switch, which is a much more affordable way to play.

GladMort is a "medieval fantasy action platformer," which means it's more or less like Ghouls & Ghosts, complete with double jumps, but it's not as punishing as the Capcom classic (however, it is challenging compared to most modern games).

The story tells of a "terrible calamity" in the kingdom, forcing the king to call upon the black knight GladMort, who isn't necessarily a good guy, but desperate times call for... a black knight. As you might imagine, the plot isn't too important to the overall gameplay, but it still provides a good backdrop for your mission as the knight.

GladMort clearly shows its retro inspiration: it clearly draws inspiration from classic console games, but it is also (very much) inspired by Capcom's Super Ghouls 'n Ghosts game for the Super NES. Although all Ghosts 'n Goblins / Ghouls 'n Ghosts games share elements in common, the elements present in GladMort seem to be more similar to those of the Super NES version. This is particularly evident in the character's double jump, which looks practically identical to Sir Arthur's double jump in the SNES game. Of course, the weapon upgrades and the way the character loses armor when hit are also clearly





derived from the latter, but the double jump is the main difference between the Super NES game and the others in the series. I also found some similarities with Magician Lord, such as the large colorful sprites, detailed backgrounds, and animations.

Its heavy inspiration from the Capcom saga has a direct impact on its gameplay. The double jump may seem a little floaty at first, but get used to its timing, as it's part of the game's overall strategy. Jump at the wrong moment and you're sure to jump straight into an enemy or land on top of them. Overall, given the homage around which it was designed, the game flows well, allowing itself to be pleasantly "tamed" while incorporating its own unique elements; it feels familiar and innovative at the same time. It can also be quite difficult in places by modern standards, although it's nowhere near as difficult as the title it draws inspiration from. Players looking for a difficulty similar to Super Ghouls 'N Ghosts may be disappointed, as GladMort is nowhere near that difficult, but modern players will probably find it challenging enough to be interesting. The graphics are excellent. Well-designed and animated characters move against colorful backgrounds that fit perfectly with the NEOGEO style and game setting. The same can be said of the beautiful soundtrack, which takes us back, in a good way, to the golden age of 90s video games.

Overall, my experience with GladMort has been positive. I had to remind myself what some of the other NEOGEO games were like, and GladMort perfectly embodies the console's aesthetic. In fact, I appreciate that the game is quite challenging, but not as difficult as Super Ghouls 'n Ghosts. I'm sure some hardcore gamers will be put off by its lack of extreme difficulty, but if you're a semi-casual gamer who likes the Capcom saga, you might really enjoy this game.

by **Roberto Del Mar Pirazzini**

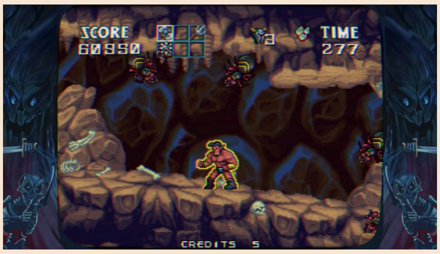
OUR FINAL SCORE

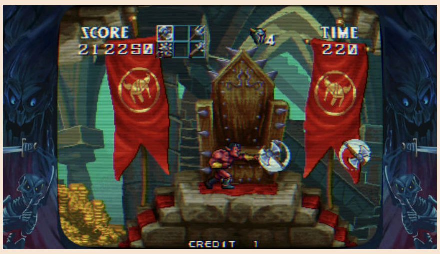
» Gameplay 85%

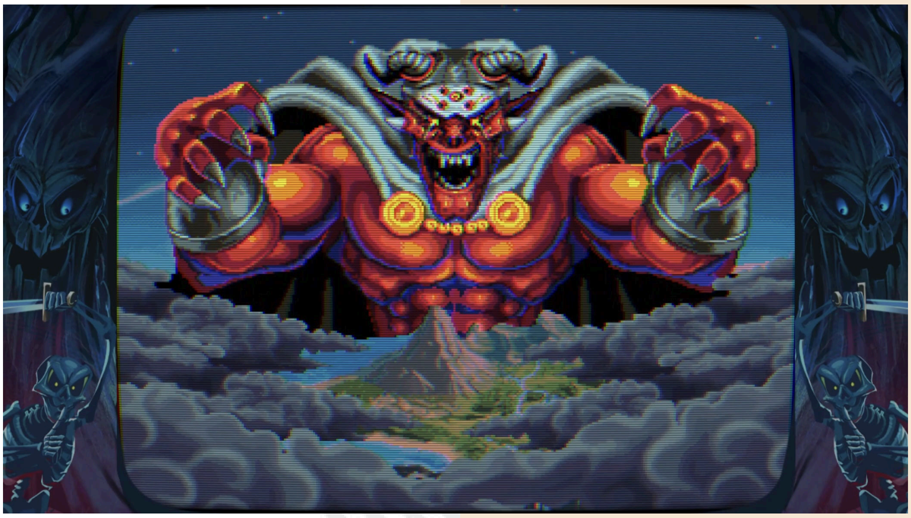
The mechanics are those of classic arcade/platform titles from the 80s/90s. The "double jump" must be managed with care.

» Longevity 75%

The mechanics are those of classic arcade/platform titles from the 80s/90s. The "double jump" must be managed with care.







NEW GAME

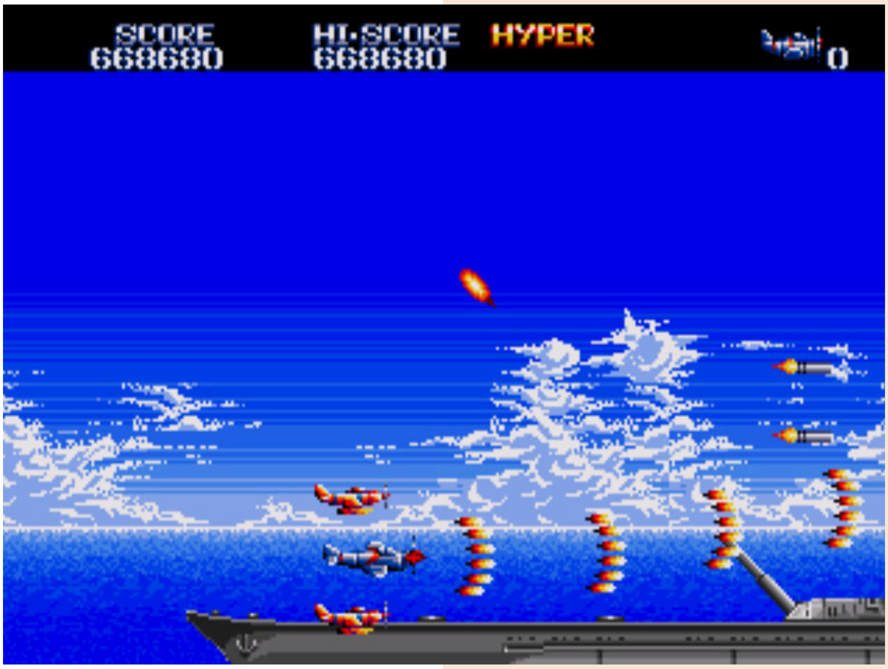
P47 II MD

Year: 2025
Editor/Developer: Retro-bit, City Connection, Jaleco
Genre: Shoot’em up
Platform: Sega Megadrive
Website: https://retro-bit.com/p47iimd/

A title with thirty-five years of history. The original P-47: The Phantom Fighter proved to be a big hit in arcades back in 1988. Several ports were released for numerous gaming systems: Commodore 64, ZX Spectrum, Amiga, and Atari ST.

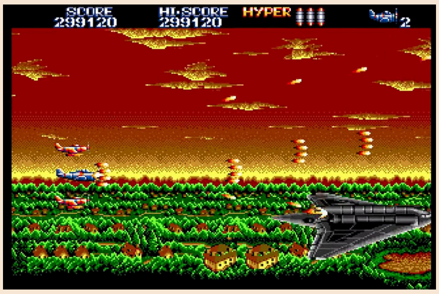
In Japan, P-47 II: The Freedom Star, Jaleco's first title for the console, was scheduled for release. Unfortunately, for various internal reasons, it was canceled.

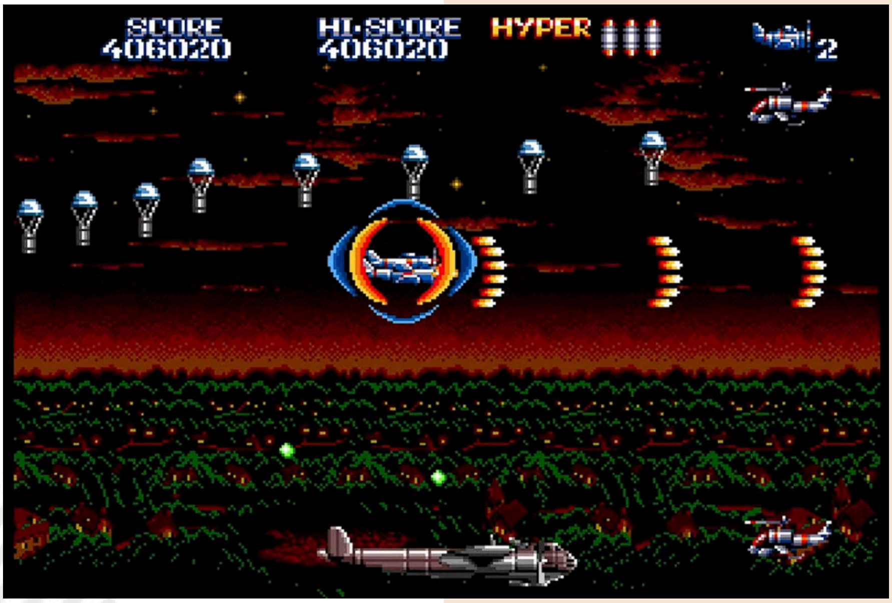
After all this time, we can enjoy the game in its entirety thanks to City Connection and Retro-bit.

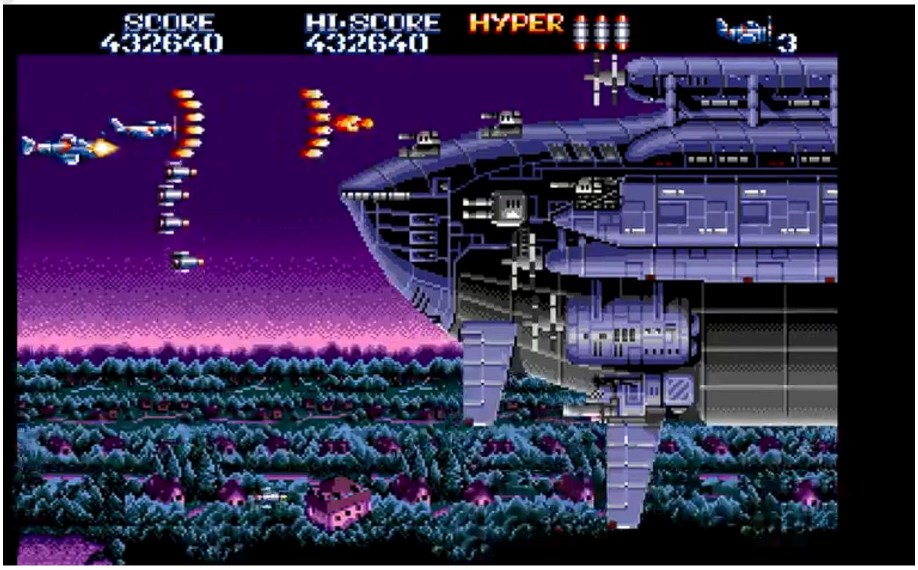


The game is set during World War II. We will be flying a Republic P-47 Thunderbolt, known as the "Jug." It was a fighter-bomber of considerable size and weight and the largest and heaviest single-engine aircraft ever built before and during the conflict (a feature not appreciated by pilots). It was famous for its poor acceleration, low directional stability, and, at low altitudes, it was reluctant to perform aerobatic maneuvers. However, it was equipped with a very powerful engine and, at high altitudes, it was very fast. It was heavily armed, robust, and had a notable feature: the absence of a vulnerable liquid cooling system for the engine (the cylinders were cooled by air), which allowed it to return to base even when badly damaged. From early 1943, it was deployed on all fronts and helped consolidate Allied supremacy in the skies during the last two years of the war.

In the game, our P-47 starts out rather weak, and we will need to recover various upgrades by shooting down enemy helicopters that carry them. These upgrades include speed







(essential, because our plane is damn slow at takeoff) and a primary fire upgrade (the machine gun), while special weapons include bombs, homing missiles, double shots, and a truly deadly shield/weapon. Each upgrade can be increased by collecting the same upgrade multiple times.

There are also "Hyper" bombs available that clear the screen, but they will cost us one of our plane's three life points, so they should be used sparingly and with skill.

The action takes place across six levels, complete with a mega boss battle finale. Among the bosses, we have the honor of facing off against the famous Japanese battleship Yamato.

Without a doubt, the boss fights are the game's strong point: they are challenging and rather well designed. They definitely increase the playability and the desire to continue level by level. In fact, the entire game is a tough test for hardcore shooter lovers, something for which the entire P-47 series is famous.

Although it can be considered a "new title," P-47 is a game from the early days of the Megadrive, and it shows. Don't expect the graphic style of Thunderforce IV, Zero Wing, or Gleylance. The graphics are beautiful and colorful, the bosses are impressive and of decent size, and the parallax is quite enjoyable, but the sprites are rather basic and there is little variety. Unfortunately, the two-player mode found in the arcade is absent.

The soundtrack is interesting and well-chosen, with a nice selection of thematic music for each level and excellent sound effects.

Compared to the best shooters seen on the 16-bit Sega, P-47 II does not stand out in the library of this genre on the console.

The Megadrive is literally flooded with this type of game, and it is necessary to offer something exceptional to stand out. Nevertheless, there is a fascinating story behind this title and a good overall production, and there are certainly worse shoot 'em ups on the console.

The cost of the boxed version is quite high, starting at $54.99 for the basic version. Each version costs an additional $10 and offers something extra in terms of packaging and included gadgets.

by **Carlo Nithaiah Del Mar Pirazzini**

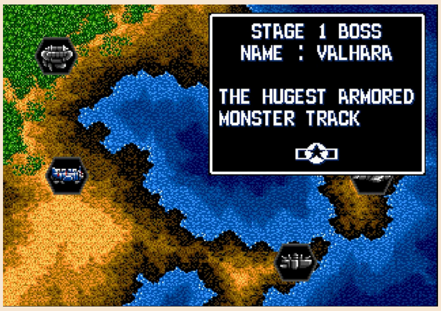
OUR FINAL SCORE

» Gameplay 85%

Enjoyable with a good control and upgrade system. Boss fights are well structured and challenging.

» Longevity 80%

It's a shame there's no two-player mode.

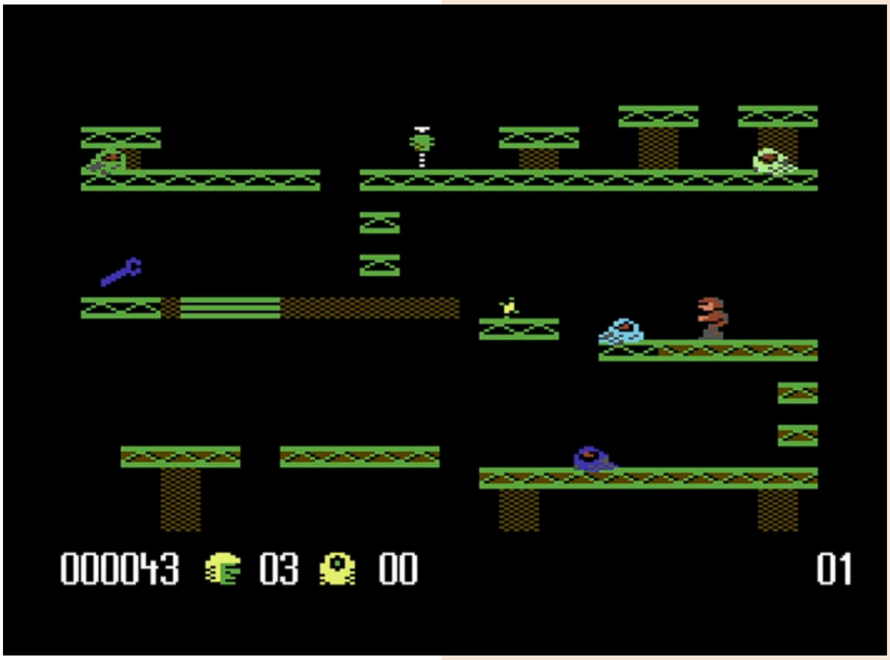


NEW GAME

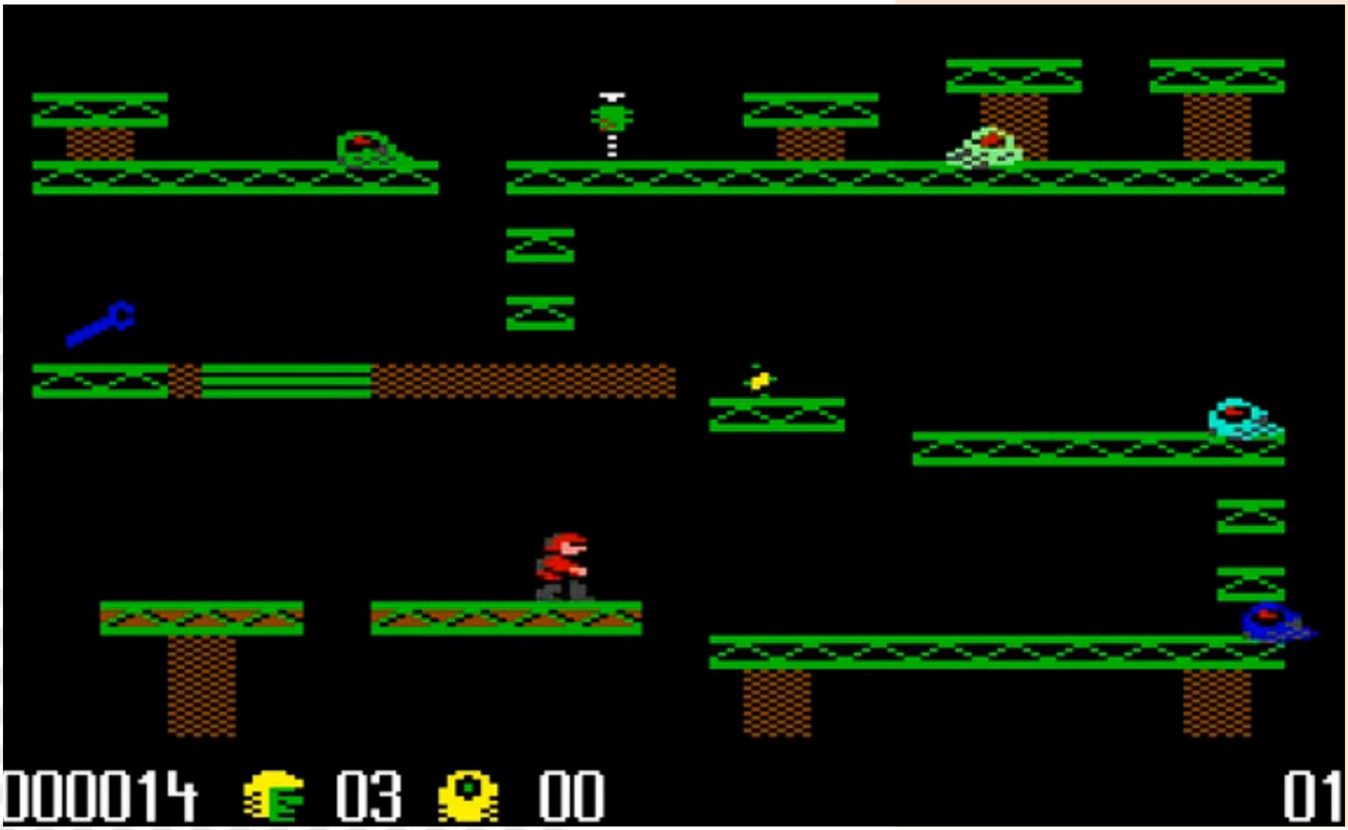
RUST ‘N’ STEEL

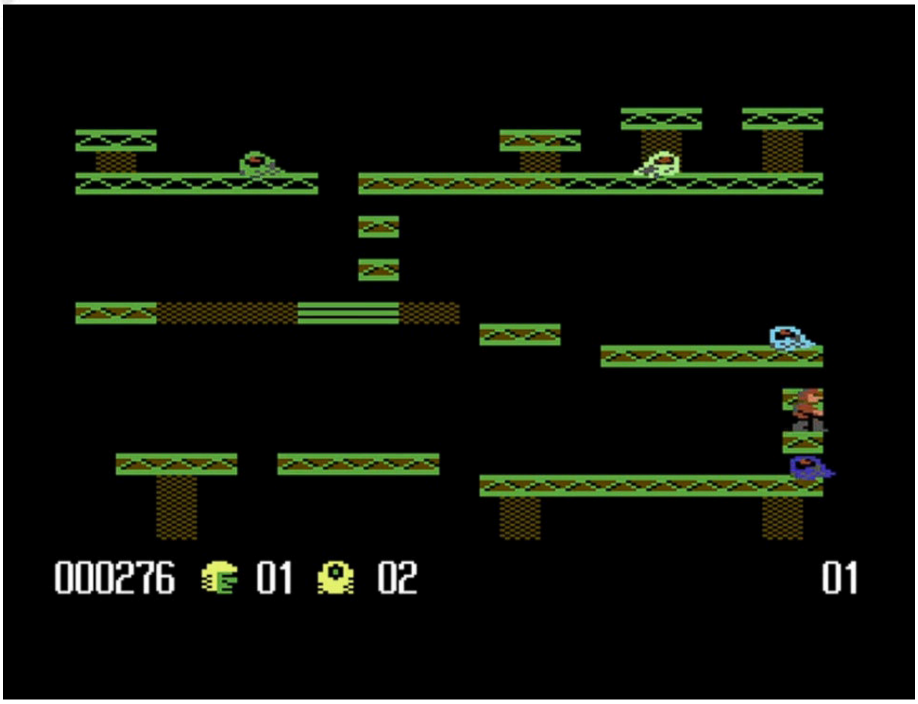
Year: 2015
Editor/Developer: natthrafn
Genre: Platform
Platform: Commodore 64
Website: https://natthrafn.itch.io/rustnsteel

If you are looking for a multi-screen platform that brings back those great Commodore games from the 80s, RUST 'N' STEEL is the one for you! This homebrew game, released in 2015, is very similar to the games that were found in cassette boxes at newsstands in the 80s and 90s. You play as an engineer who has to jump on steel platforms, turning them into rust, while avoiding various enemies roaming around the screen and disastrous falls that will turn you into a pile of bones.



The game is fun, but in my opinion it has some significant flaws: the main sprite's movement is rather wooden, making it difficult to jump from one platform to another. This makes the gaming experience a tad frustrating





OUR FINAL SCORE

» Gameplay 65%

It's difficult to give a higher rating. The protagonist's movement greatly ruins the gameplay, leading to intense frustration right from the start.

» Longevity 67%

Again, it's a shame! Really! Because the game idea is interesting and the levels are well structured. More could have been done.

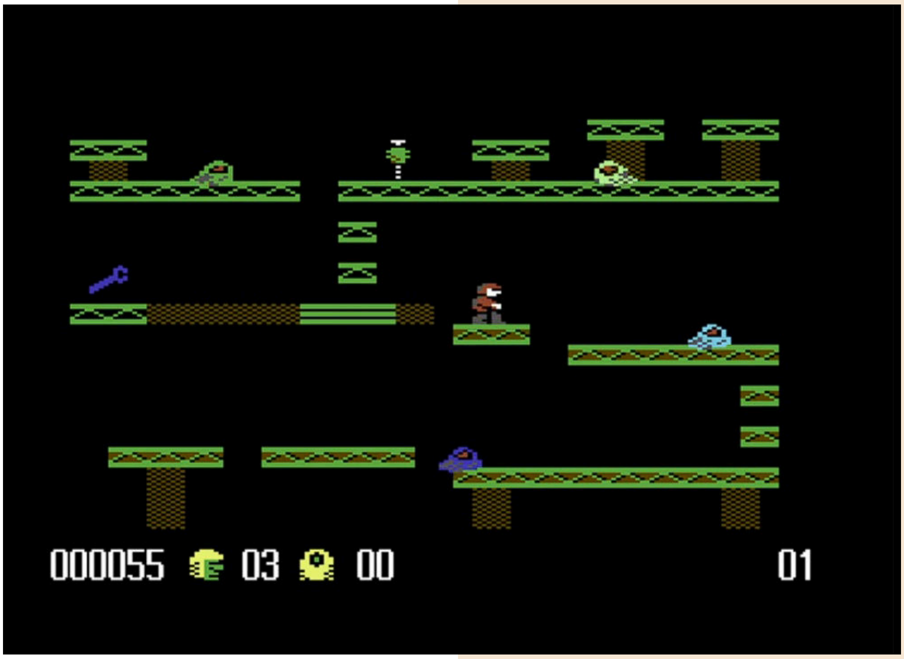
and risks tiring you out almost immediately, with the risk of a "Hard Reset" even after just 10 minutes of play. The use of the color palette is not great. I've seen better on the C64, and they could definitely have done a better job. It's "gloomy" and, combined with "annoying" sound effects, does nothing to help create atmosphere.

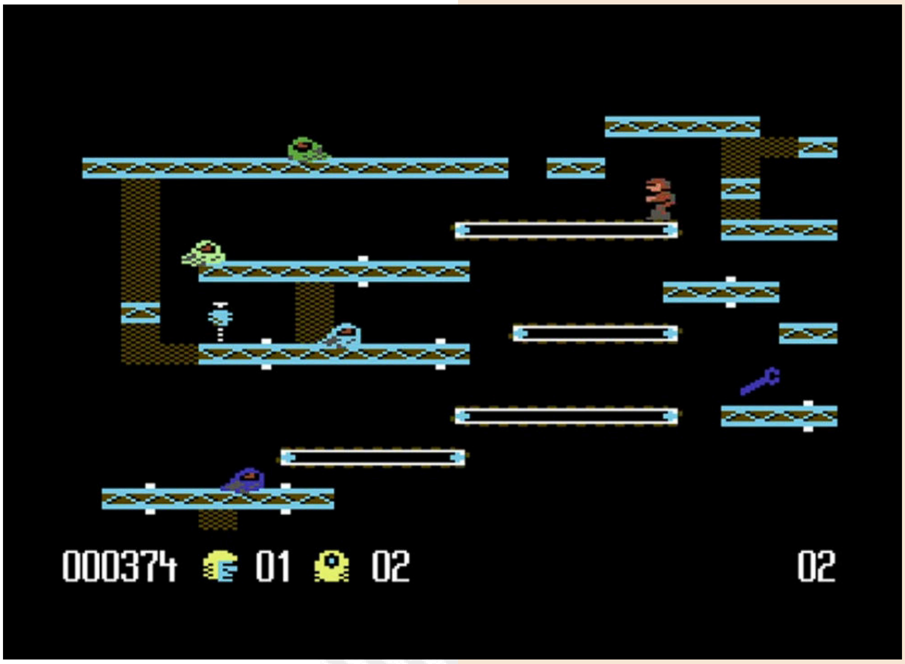
An interesting but decidedly mediocre game, which might have had a better chance if it had been released in the mid-1980s rather than recently. In 2015, they could have done better!

It's a shame, because the idea behind Rust 'n' Steel is interesting and the type of game is always enjoyable.

In short, in my opinion, this is a mediocre game. It would certainly have been a good game if it had been released in the 1980s, but as a 2015 Hombrew, it could have been a little better.

by **Chiara "Kika" Carrara**





BATEN KAITOS:

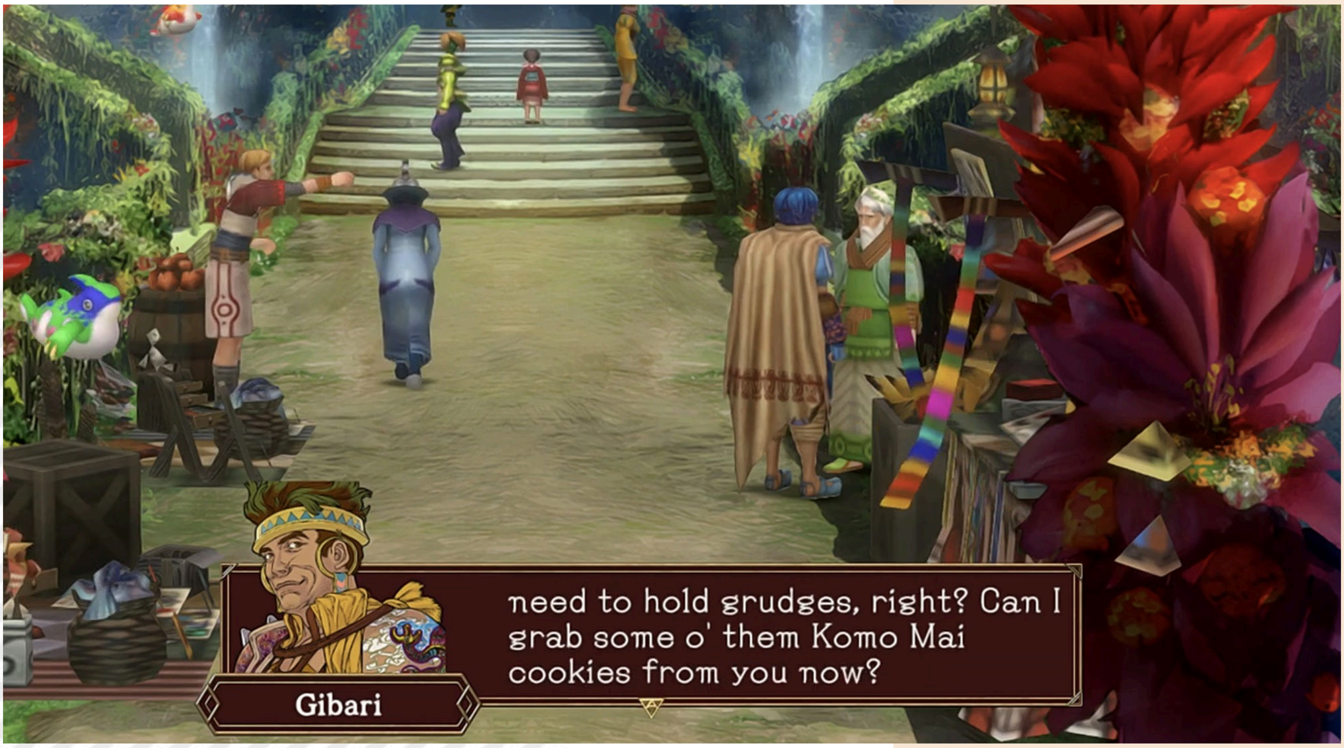
ETERNAL WINGS AND THE LOST OCEAN

Year: 2003
Editor/Developer: Monolith Soft., tri-Crescendo Inc.
Genre: Jrpg
Platform: Nintendo GameCube, Nintendo Switch
Website: https://it.bandainamcoent.eu/baten-kaitos/baten-kaitos-i-ii-hd-remaster

Personally, I find that one of the greatest strengths of the Nintendo Switch is the huge number of high-quality remasters and ports of titles from the golden age of PS1/PS2 and GameCube. Although some well-known titles are missing (e.g., Skies of Arcadia), for JRPG fans, these re-releases represent a sort of neoclassical revisiting of the genre.

Baten Kaitos I & II HD Remaster contains the first title, "Baten Kaitos: Eternal Wings and the Lost Ocean," and the second, "Baten Kaitos Origins." While Origins is a prequel that was never released in Europe, Eternal Wings was also distributed in Europe and contains texts in various languages, including Italian.
A notable difference from their original edition is the lack of English dubbing, which, despite being unpopular with players at the time, I would have personally preferred to the Japanese dubbing, as not being familiar with this language hinders my involvement.
In addition to various graphical enhancements, these remasters also offer modern features already seen in other re-releases of SquareEnix titles: "quality of life" features that almost seem like cheats, such as the ability to skip encounters or make battles automatic (you might as well not play a JRPG).

Eternal Wings is a very innovative game in terms of ideas. In addition to featuring a setting composed of flying islands (a taste of what Xenoblade 2 will be), its combat system and many elements of the story are based on playing cards called "Magnus" used daily by the islanders to contain and transport objects, food, weapons (in Origins even emotions).
In addition to the Magnus that will be used outside of combat to transport mission items needed to solve puzzles in dungeons or side quests, our characters will have a deck of Magnus to customize and use in combat. This contains their equipment, skills, spells, relics, and other truly unique items to use in battle, such as eggs, seeds, fruit, or even paintings. Another unique aspect is the developers' use of the passage of time to make Magnus "evolve." In fact, despite being inside a card, items are not immune to the passage of time: in an hour of



need to hold grudges, right? Can I grab some o' them Komo Mai cookies from you now?

gameplay, fruit could rot, eggs could hatch, or milk could turn into yogurt. This change alters the characteristics of Magnus, which can become more powerful or weaker.

An interesting gameplay aspect of Magnus is the ability to create new ones using specific combos in battle (SP Combos). For example, casting wine and fire magic in the same turn will create the new Magnus "warmed wine."

Finally, combat is divided into attack and defense turns. We will have to play offensive cards (weapons, magic, items that cause altered states) or defensive cards (healing, stat buffs) during our character's turn, and use defensive response cards (armor, helmets, shields) when we are attacked during the enemy's turn. These long chains of cards, which will only increase as we level up, will empty our characters' decks (even the characters' hands of cards are affected by level), which will be reshuffled, causing us to lose a turn (this also applies to enemies).

The most unique aspect of Baten Kaitos is undoubtedly its setting. The game takes place on five islands that took flight a thousand years ago following the War of the Gods. Most of the inhabitants of these islands can sprout a pair of wings called Wings of the Heart from their backs, a legacy of their ancestors, with which they can fly short distances.

Our protagonist, Kalas, is unique in that he has only one biological wing and one artificial wing, a gift from his grandfather. Following his thirst for revenge, Kalas will meet the other members of the party and, traveling with them across the various islands, will try to thwart Emperor Geldoblame's lust for power.

As players, we take on the role of an incorporeal spirit who communicates directly with Kalas. Kalas will often turn to us throughout the story, asking us questions about how to proceed or asking for our opinion. Our answers will not only lead to different dialogues with Kalas but will also influence a secret statistic that is important for the chemistry of combat.

Unfortunately, as far as I'm concerned, there isn't much else to say. The story of Baten Kaitos never really takes off and fails to give enough space to the secondary characters. This is true until a certain point in the story, when what is probably one of the best plot twists in the history of the genre is pulled out of the hat, but unfortunately it has a resolution that is too convenient, leading to a very tired final act that drags on until a bittersweet ending.

Fortunately, the party characters are given some development in their side quests. These are only accessible in the last part of the game, but they make up for narrative shortcomings that would otherwise have been unforgivable, especially considering that we are talking about the people responsible for titles such as Xenogears and Xenoblade Chronicles.

Furthermore, Baten Kaitos Origins, being a prequel, greatly enhances some of the party members and even some of the antagonists, giving Eternal Wings a whole new depth. In fact, I would personally recommend playing Origins first.

OUR FINAL SCORE

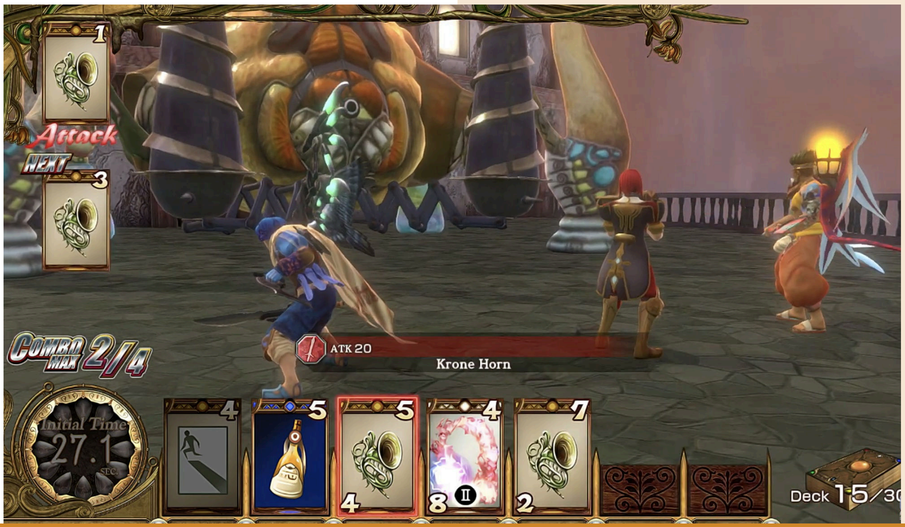
» Gameplay 75%

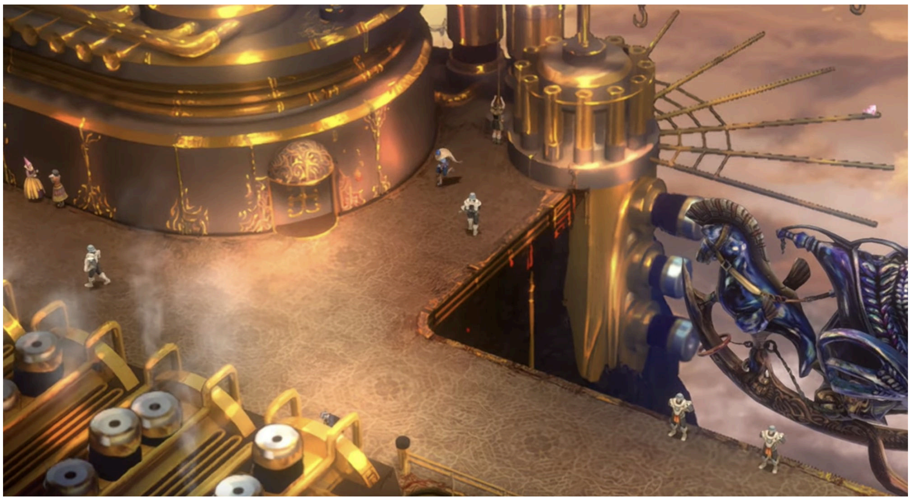
As a JRPG of its time, it still holds up well today. All the improvements that could be made to the Magnus system will be applied in Origins. From a narrative point of view, some choices have been made that kill the pace of the game and are sometimes frustrating. Some dungeons are not exactly the best.

» Longevity 90%

Personally, I am against JRPGs that last 100+ hours. I find that the right compromise for the story to remain engaging is between 35 and 50 hours maximum. Baten Kaitos falls into this category by focusing solely on the main story. However, there are many achievements available, such as the ability to photograph all monsters, collect all Magnus, or unlock all SP Combos, so there is something for even the most die-hard perfectionists (good luck without a guide).

The love story between the two main characters is also very forced and cobbled together, one-sided for three-





quarters of the game and then becoming central without the necessary build-up and motivation. Final Fantasy VIII and Final Fantasy X remain the masters in developing a love story that is the centerpiece of the entire game.

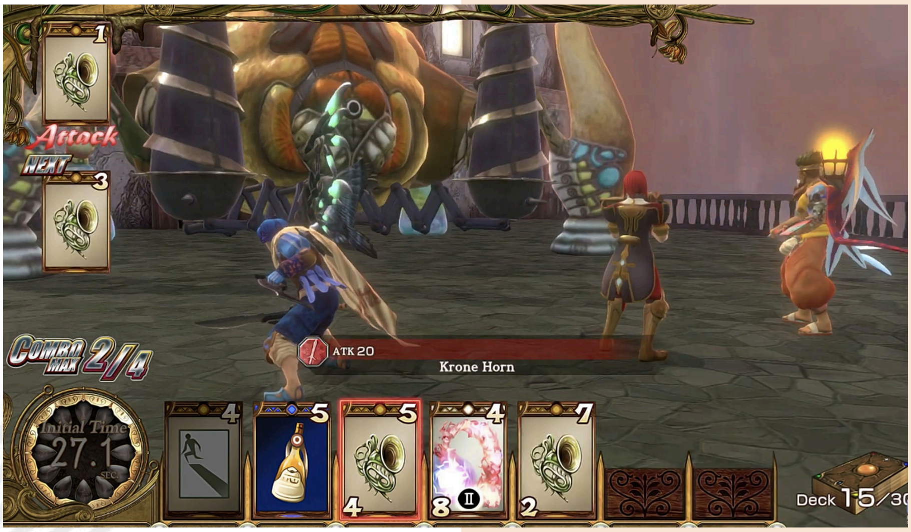
The imagery of the islands we visit during our travels is dreamlike. Baten Kaitos offers dreamlike landscapes reminiscent of paintings: from beautiful jungles and forests to the completely surreal scenery of Mira Island, then on to the steampunk brutalism of Mintaka. Even the inhabitants' clothing, furnishings, customs, and traditions are very varied and imaginative. From this point of view, the game does an impeccable and memorable job.

The music is also very beautiful and well-designed, although for such a long game there is very little of it and it tends to repeat itself. Even very catchy tracks risk becoming a torment (right, Anuenue?). Some of these, as if that weren't enough, can even be found in Origins.

The dungeons are hit and miss. Some feature frustrating puzzles, while others are more of a passing through. However, they are all very beautiful to look at and explore.

The collect-a-thon-style side quests are excellent. Both the family tree and the animals and constellations are very satisfying to complete.

One negative note is the game's economic system. The only way to really make money during the game is to photograph monsters and bosses and then sell the photos to shopkeepers (after some time has passed and the game has developed), which requires the use of the Magnus camera during an attack in battle. The problem, however, is that at the end of the battle, we can only keep one item from the loot left by the monster, photos, and Magnus generated by SP Combos. This unfortunate choice often forces us to opt for a rare item rather than the photo. This forces us to fight the same monster again to photograph it again in order to earn money and unlock the related achievement.



In addition, there are some very questionable gameplay choices, such as forced backtracking and bosses that must be faced several times in a row without any real motivation or even changes in strategy in some cases, choices that greatly disrupt the rhythm.

Being a game published by the then Namco, there are also several very entertaining Pac-Man-themed Easter eggs.

In conclusion, I found "Baten Kaitos Eternal Wings" to be a decent JRPG, with excellent ideas but many shortcomings in terms of story and character development. This is the exact opposite of Baten Kaitos Origins, where I hated the combat but loved the story and characters. However, both games have a wonderful imagination going for them.

If your favorite part of a JRPG is the narrative, I recommend playing Origins first and then moving on to Eternal Wings. Both the story and the setting will benefit greatly from this.

by **Pietro “WarioPunk” Turri**

NEW GAME

BONI IN BAT HELL!

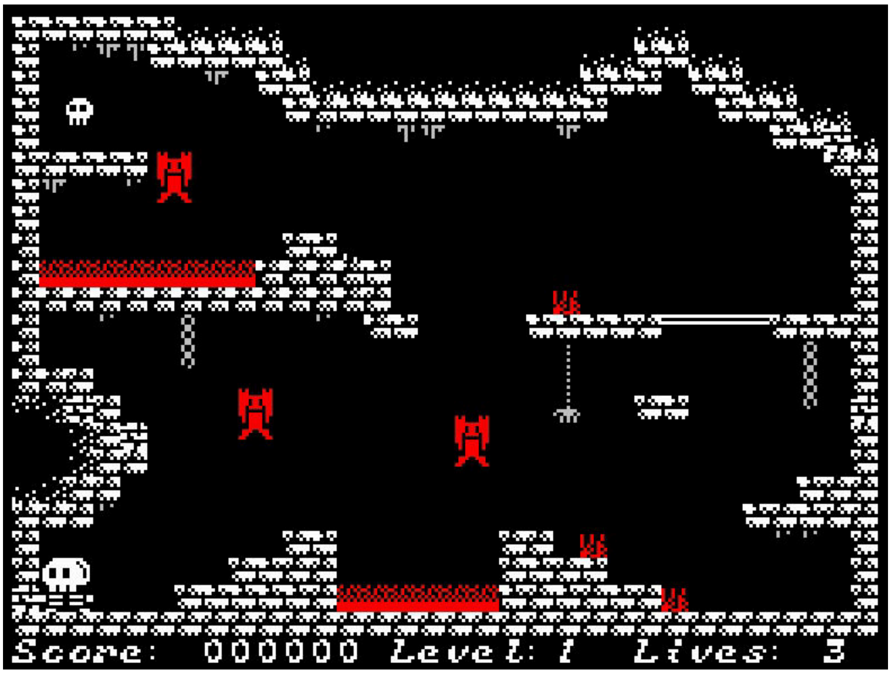
Year: 2025

Editor/Developer: SteveTyson

Genre: Platform

Platform: ZX Spectrum

Website: https://stevetyson.itch.io/boni



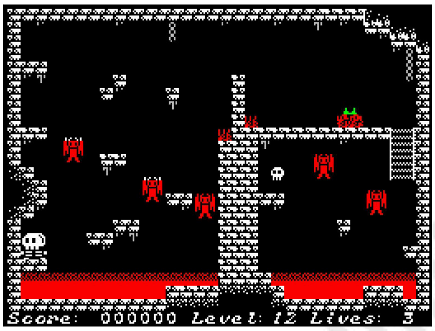
Hellish in both title and gameplay, this is Boni in Bat Hell! for ZX. A platform game born from the sadistic mind of Steve Tyson. Don't be fooled by the cute graphics... this game is devastating, difficult, and distressing. It's one of those titles created with the intent to torture. The goal is to collect small skulls in 20 levels to bring our little (undead) friend back to life and get him out of this absurd hell populated by bats of various colors. The game is a classic fixed-screen platformer. You jump, reach the various platforms, collect extra lives, and try to get to the skull... but the problem is the monstrous difficulty that the developer has thrown at us.

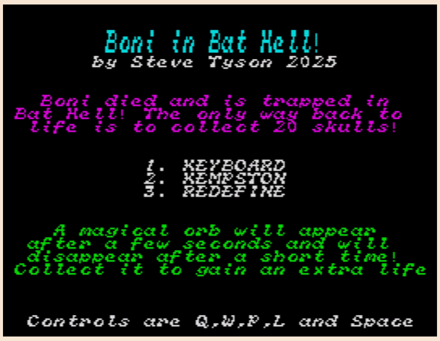
Everything you do must be "pixel perfect": every movement, every move, every jump, every calibration. Deadly, I assure you. A test of courage and reflexes that I haven't seen in a long time.

Maybe it's because old age has slowed my reflexes, but this is beyond the limit. It's a real shame because the game is aesthetically very cute and has beautiful music (even if it does become repetitive after a while).

I don't feel like rejecting it, but I certainly don't recommend it to those looking for a platform game with gradual difficulty.

by **Giampaolo Moraschi**





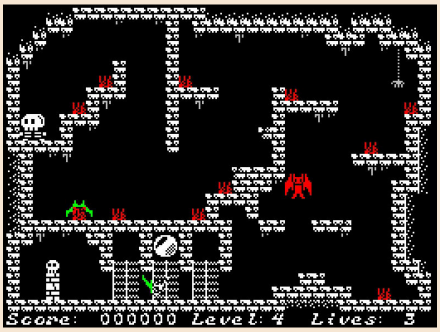
Boni in Bat Hell!
by Steve Tyson 2025

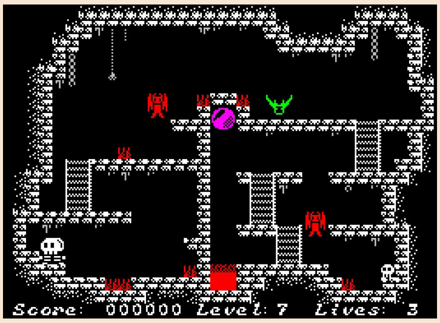
Boni died and is trapped in Bat Hell! The only way back to life is to collect 20 skulls!

1. KEYBOARD
2. KEMPSTON
3. REDEFINE

A magical orb will appear after a few seconds and will disappear after a short time! Collect it to gain an extra life

Controls are Q,W,P,L and Space





OUR FINAL SCORE

» Gameplay 65%

Too imprecise in some commands and with very irritating gameplay.

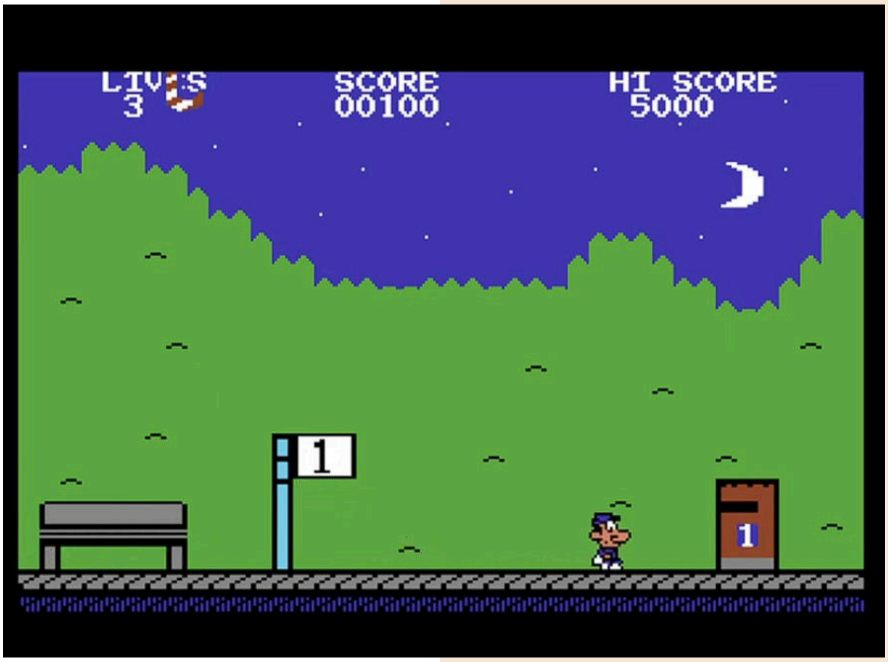
» Longevity 60%

If you're a masochist, raise the score by 30 points, but here we're talking about pure delirium. Useful for learning new swear words.

NEW GAME

EL CARTERO

Year: 2025
Editor/Developer: Bonnette2020
Genre: Platform
Platform: Commodore 64
Website: https://bonnette2020.itch.io/el-cartero-c64-demo



If you thought that the postman's job was a quiet stroll among terraced houses and smiling old ladies, El Cartero will make you change your mind faster than a lost registered letter! In this little homebrew for Commodore 64, the postman becomes a sort of urban hero, armed only with an empty mailbag and lots and lots of patience!

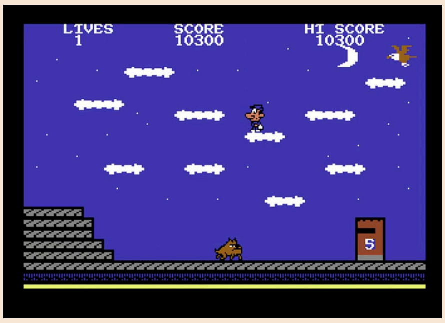
Welcome to El Cartero, a new title for C64 where our hero doesn't deliver mail, but runs like crazy to collect it, dodging obstacles and trouble in a city gone mad. Why? Who knows. Maybe the post office lost the shipment? Maybe there was a strike? Maybe our Cartero is just too enthusiastic about his job.

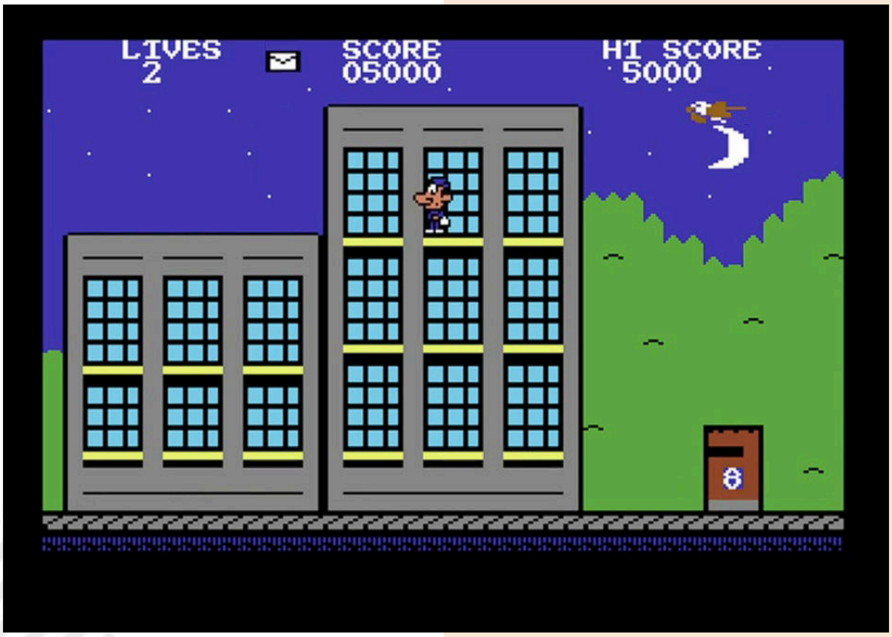
Through various levels around the city, our hero must collect 9 letters falling from the sky as quickly as possible, avoiding stray dogs, birds pooping in flight, and many other strange and bizarre enemies.

During the game, you can also collect some candies that will increase your final score, in pure arcade style.

The game features 14 levels in total and, despite its "simple" appearance, will drive you crazy with its fast pace and great fun.

It makes wonderful use of the color palette of our beloved C64. The graphics are colorful and well-finished, and the protagonist is well animated! The levels are very varied, encouraging the player to discover more and more of what lies ahead as they continue with the "hard work of the postman." The soundtrack is also cute and





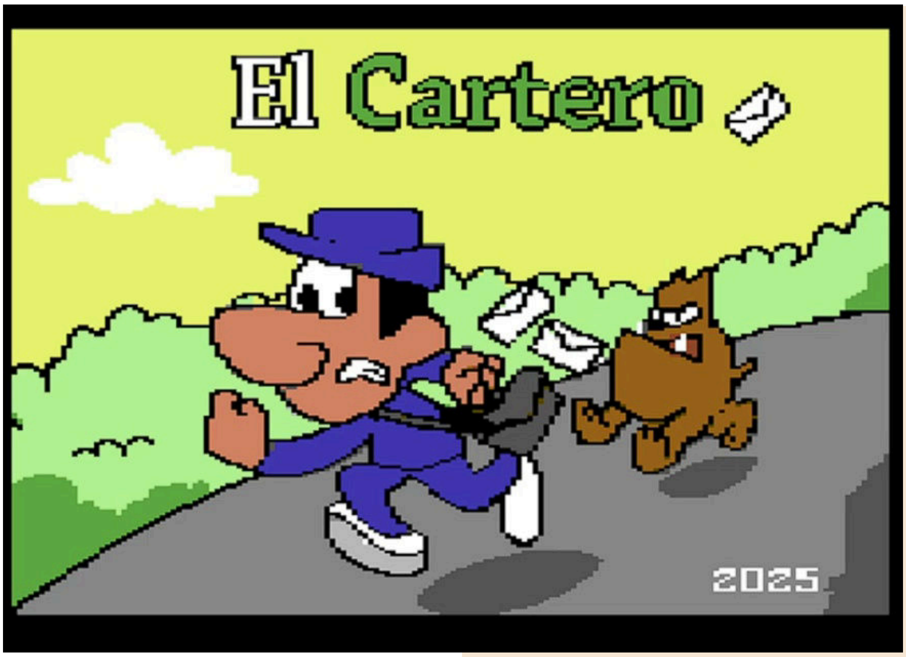
cheerful, making good use of the SID.

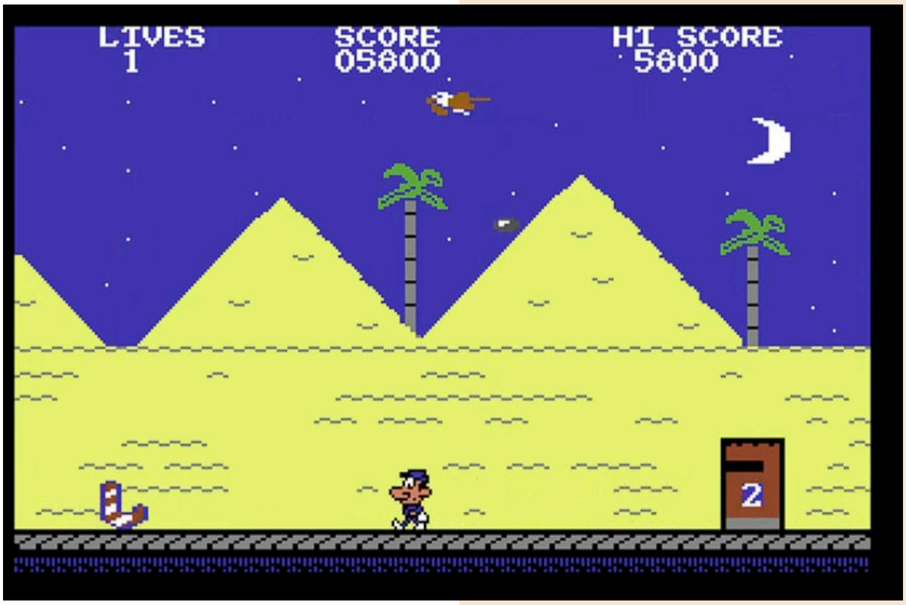
The only flaw in the game is the excessive repetitiveness of the action (personally, it didn't bother me and I particularly enjoyed it). In any case, it's a simple and enjoyable title and... if we're being honest, in the 1980s we played much worse games on our "big cookie."

The game was developed by AdriÃ¡n Bonnette, Dirk Roosen, and Esteban Trujillo with the new C64 Game Maker 2.1.1 and is available in physical format on both cassette and disk.

To sum up, I think the game is perfectly adequate. In fact, it's even more than that because it's fun and beautiful to look at. Now all you have to do is turn on your Commodore and try it, because... as I always say... there's nothing better than TESTING a game with your own joystick at home!

by **Chiara “Kika” Carrara**





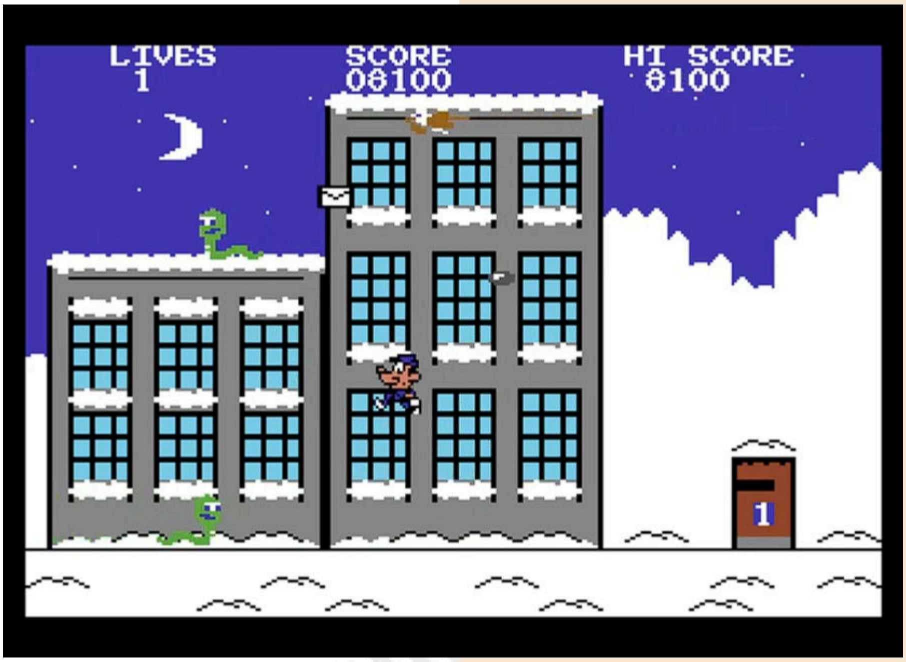
OUR FINAL SCORE

» Gameplay 90%

This "cartero" is simple but fun to play. The graphics are well done and the sprites are very cute! Definitely worth trying!

» Longevity 85%

A fun game to pass a few carefree hours. As mentioned, it becomes a tad repetitive in the long run, but the screens are always different and curiosity leads you to play it again and again and again.



NEW GAME

TRACK ‘N’ FIELD

Year: 2025
Editor/Developer: JOTD
Genre: Sportivo
Platform: Amiga AGA
Website: https://jotd666.itch.io/tracknfield

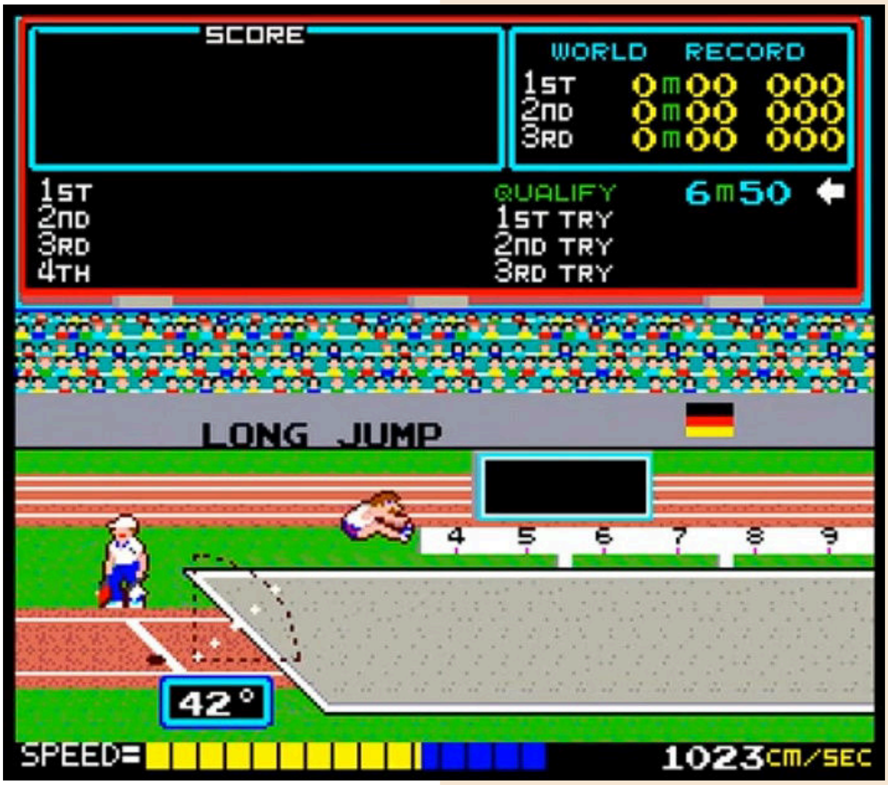
Developer JOTD is more active than ever on Amiga. After delighting us with Gyruss, it now brings us this conversion of Konami's classic sports game.

First released by Konami in 1983 for arcades and later converted for a myriad of different platforms, Track 'n' Field, as the name suggests, is a sports game based on track and field events. One of the most famous conversions was the NES version, released in 1987 and then re-released in 1992 with a version dedicated to the Barcelona '92 Olympic Games.

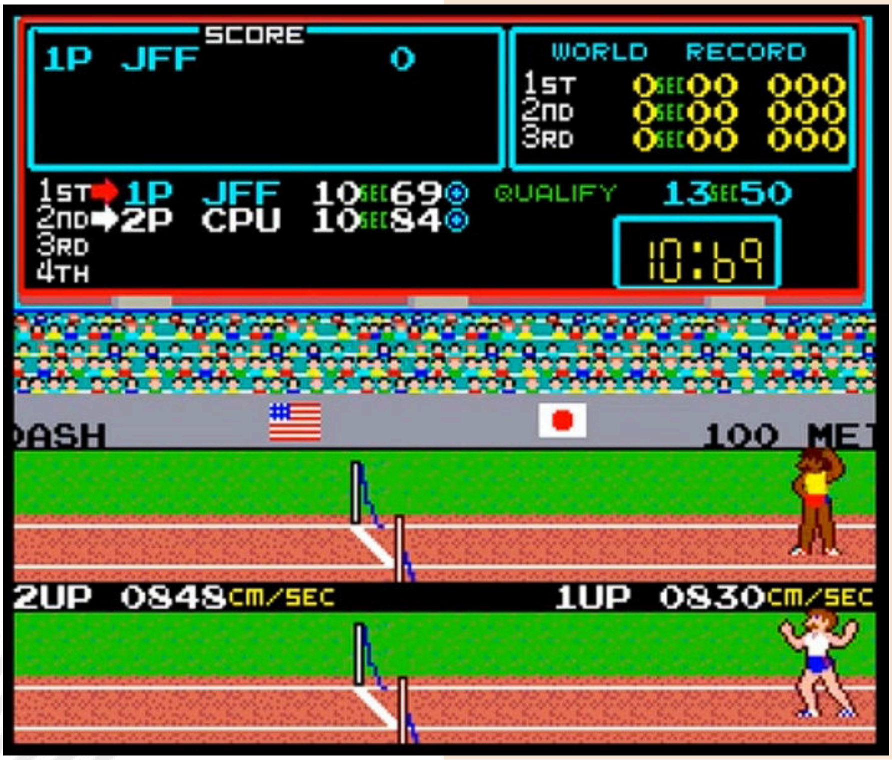
The gameplay is generally very simple: you participate in numerous sporting events by quickly pressing the game buttons. In fact, the original arcade cabinet did not have a joystick. However, this simplicity was a real 'massacre' for the hands and controllers in the numerous home versions.

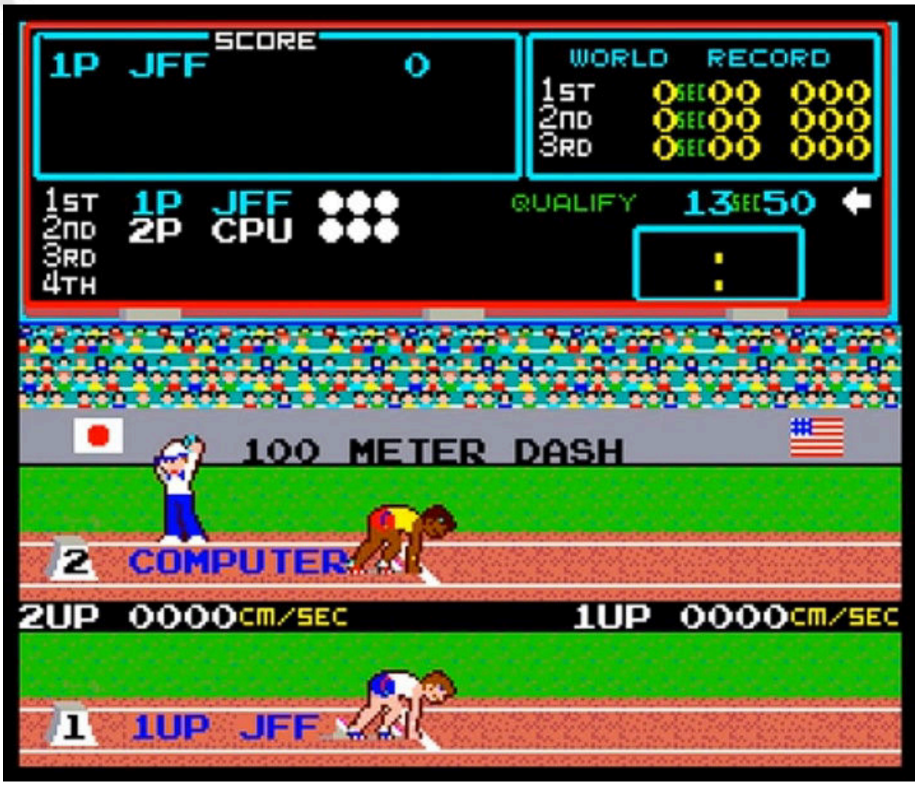
The first discipline is the 'queen' of the Olympics, the 100-meter dash. You have to press the buttons furiously and quickly to finish the race in the allotted time and maybe even set a record. Next is the long jump. More ultra-violence on the buttons, enough to cause blisters and numerous curses to various deities. Then came the javelin throw, the 110 hurdles (beautiful), the hammer throw, and finally the high jump. Needless to say, speed, rhythm, and power on the buttons are also key in these disciplines.

In short, it's a highly competitive game. Both with yourself and with friends.



After years, we are seeing a noteworthy Amiga version, faithful to the arcade





OUR FINAL SCORE

» Gameplay 80%

The archaic but functional game system is still the same as the one seen in arcades. Your joysticks/joypads will be put to the test.

» Longevity 75%

You have to like the arcade title. It's a title that shows its age and the game system. I appreciate it, but I'm convinced that not everyone is a fan of Konami games.

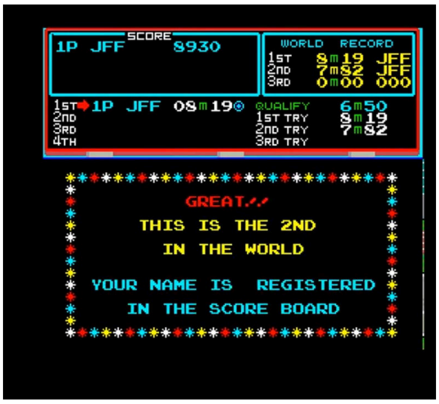
game. Indeed, as JOTD teaches us, it is a flawless piece of reverse engineering, with all the difficulties of the original game.

The title runs on all Amiga AGA computers, but an OCS version is planned. It is possible to use controllers with two buttons and it is compatible with the CD32 pad.

What more can we say? It's the arcade game, for better or worse. Enjoyable to play for the first 5 minutes, but then you find yourself shouting and getting into the classic "competitive" mood that already characterized it in the arcades.

Try it... I recommend it! And watch out for blisters!

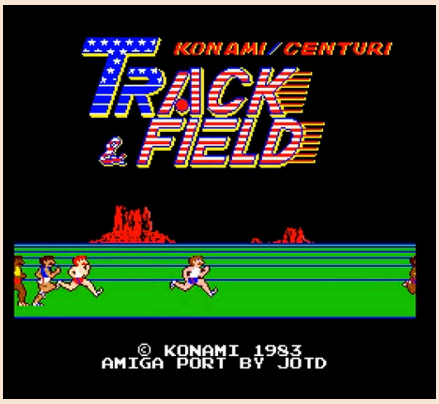
by **Giampaolo Moraschi**

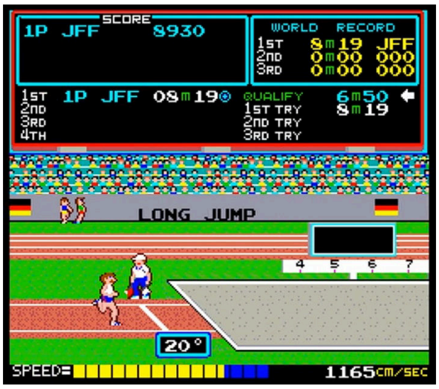


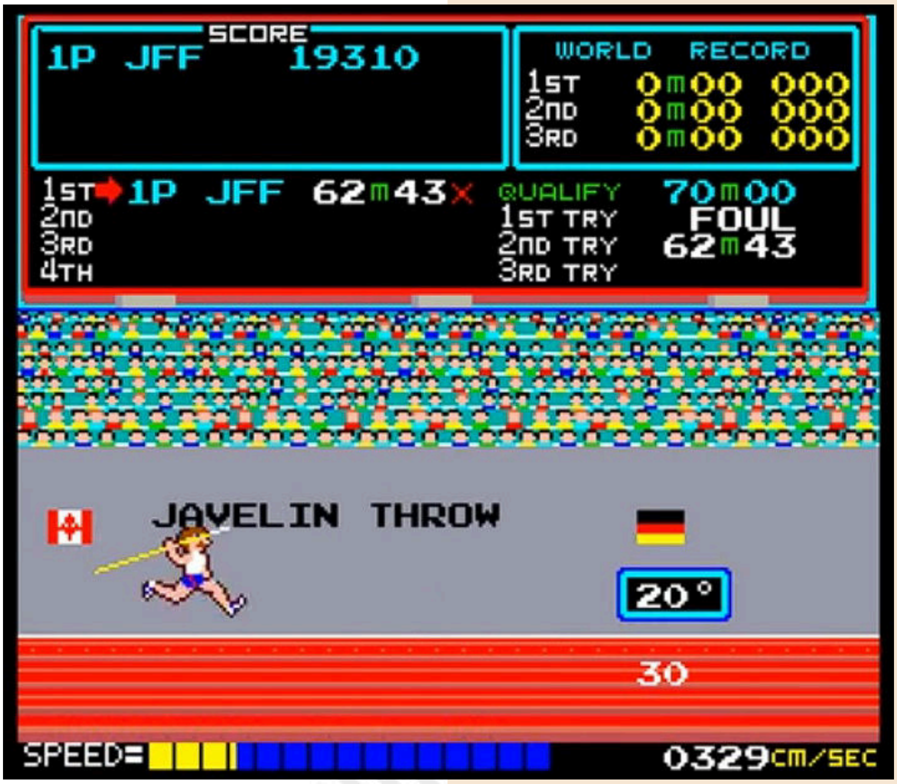
GREAT..!

THIS IS THE 2ND IN THE WORLD

YOUR NAME IS REGISTERED IN THE SCORE BOARD





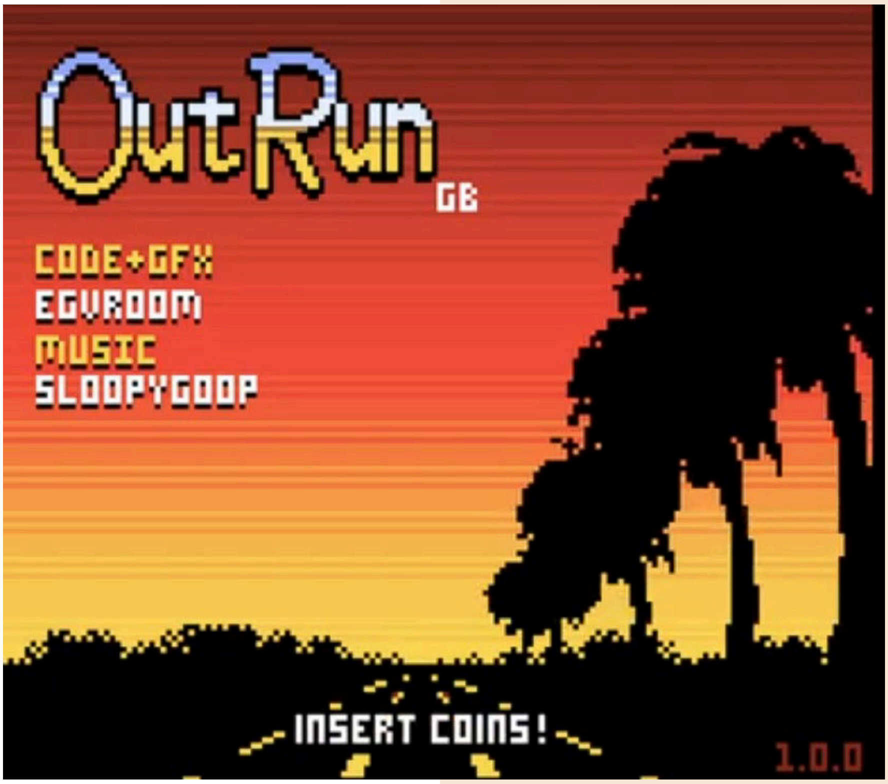


NEW GAME

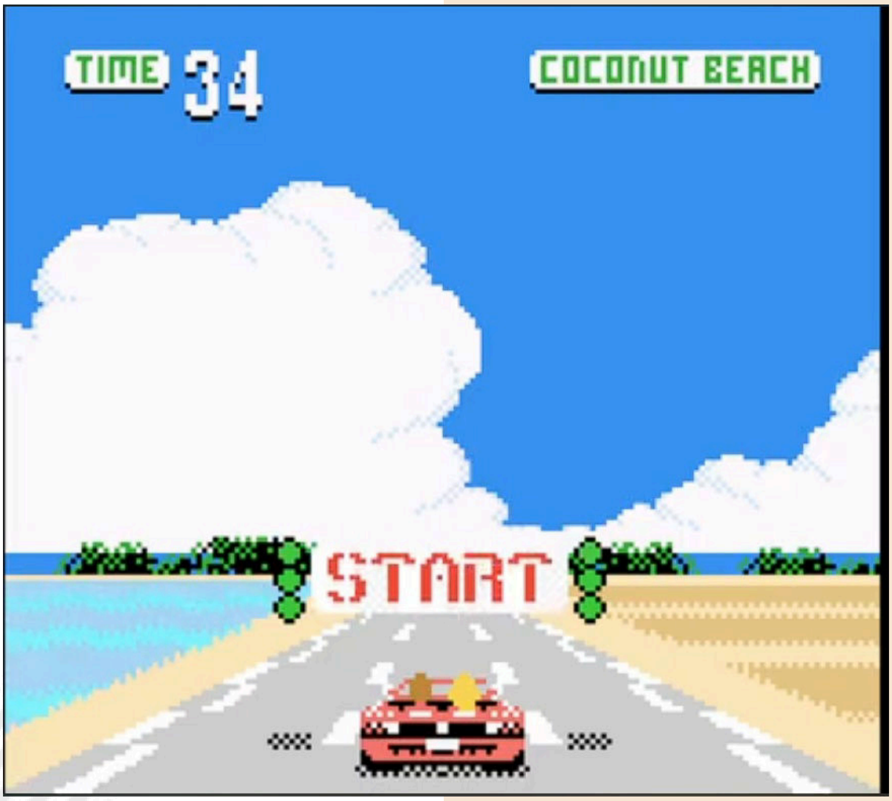
OUTRUN GAME BOY

Year: 2025
Editor/Developer: Rocketship Park
Genre: Driving game
Platforms: Game Boy Color/ Analogue Pocket
Website: https://itch.io/queue/c/439443/egvrooms-game-boy-games?game_id=3557310&password=

It's OutRun summer! After the excellent (but expensive) Amiga version, here comes this portable version of Yu Suzuki's classic.

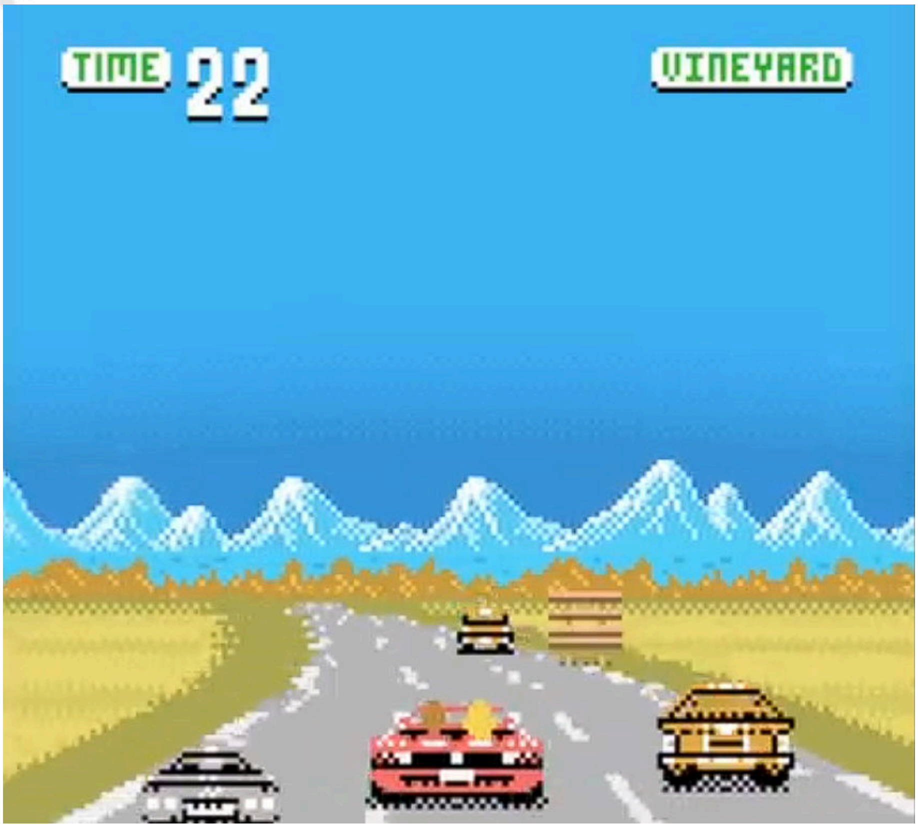


The question the developer asked himself was, "I wonder what OutRun would be like on the GB Color?" The answer is right before our eyes. A port of the famous 1986 arcade racing game in every way.
The Nintendo handheld console (both Color and Classic versions) never saw a version of the game at the time. Perhaps due to hardware limitations or, more likely, because it was a Nintendo machine and OutRun was one of rival Sega's flagship titles.
Now, in 2025, all this is possible and it has turned out well. Of course, the title is not and never will be arcade perfect, but the spirit is there.



All the game tracks are there, and you can 'drive' them with the same feeling of freedom and desire to move forward as in the arcade version. Access to the levels has been modified for memory reasons: every time you reach a checkpoint, the game restarts as if it were a new level, but the effect is the same.

Graphically, it is very well done. I notice a lot of technical work in the latest GB productions in general. Great care has been taken with the animations, use of colors, and game



OUR FINAL SCORE

» Gameplay 90%
OutRun also lives on in a portable version, thanks to a remarkable conversion and some interesting tweaks.

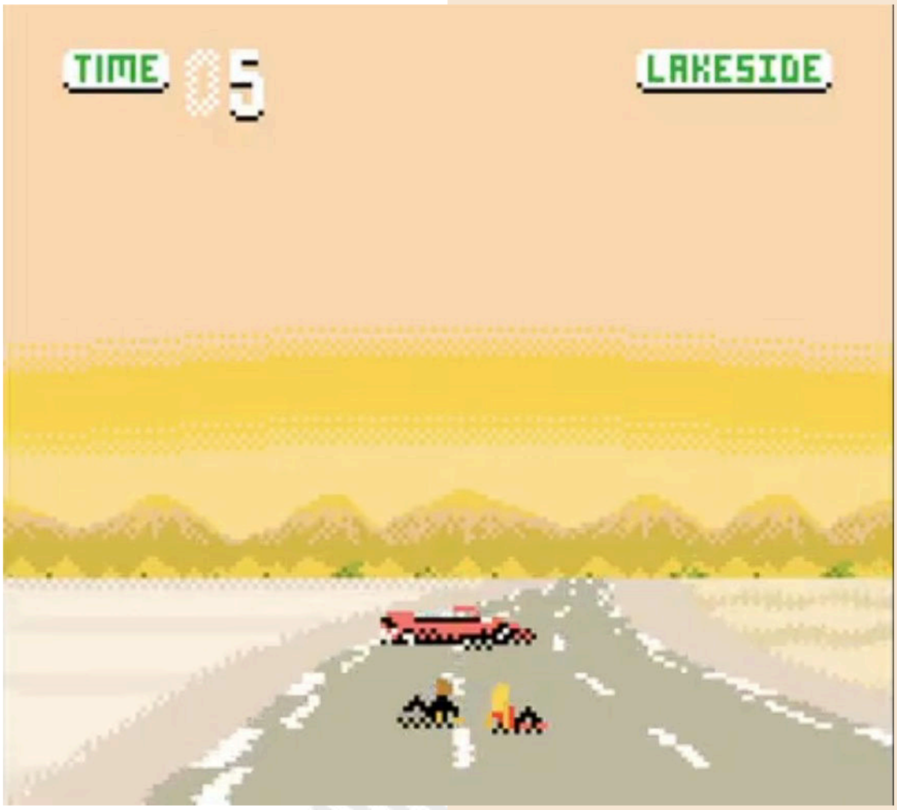
» Longevity 85%
Difficult but beautiful even on GBC.

speed.

Praise must go to Sloopygoop, our 'musician', who took Hiroshi Kaguchi's legendary soundtracks and squeezed them into the game's 512kb in a... perfect way! Well done indeed! The beautiful soundtrack is all there!
Everything else? It's just right: gameplay feel, simple controls, responsiveness, and increasing difficulty.

In short, a welcome game on GBC and also in the Analogue Pocket version, just waiting to be enjoyed and played. Have fun!

by **Giampaolo Moraschi**



NEW GAME

RANDOOM ANCIENT STONES

Year: 2025
Editor/Developer: Picaro Games
Genre: Platform/Puzzle
Platform: Commodore 64
Website: https://dozznar.itch.io/randoom-ancient-stones

Ready as ever, I'm back to review a new "gem" for our beloved C64, this time a very interesting game: Randoom ancient stone.

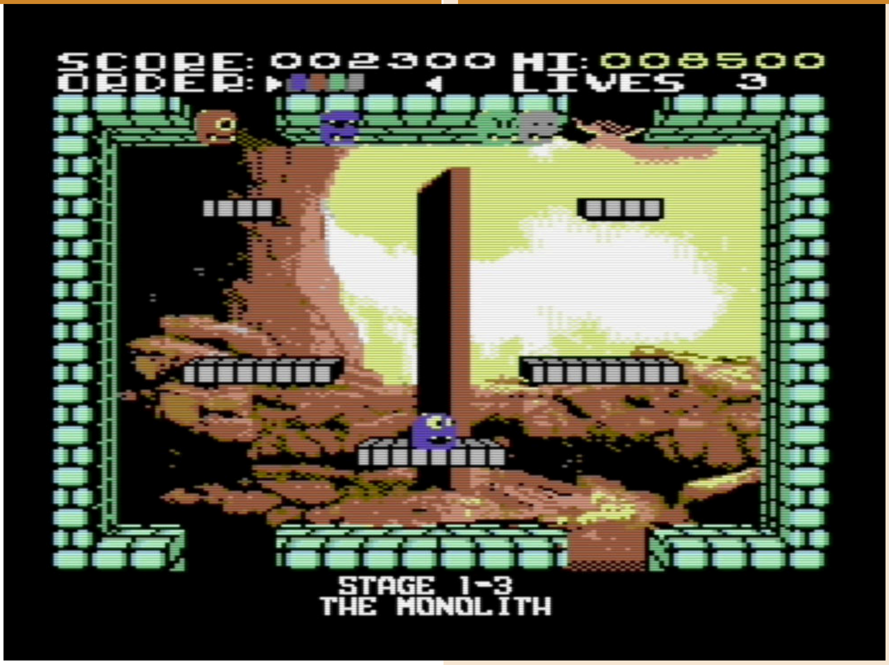
Developed by Picaro Games, Randoom Ancient Stone is an arcade game where you have to visit temples and ancient sites to recover the ten missing keys needed to complete the adventure.

Our round, gelatinous hero must absorb the colors of the enemies present in each level. The color of our character indicates the color of the enemy he must chase and capture. But if we hit an enemy with a different color, we will lose one of the three lives available to us.

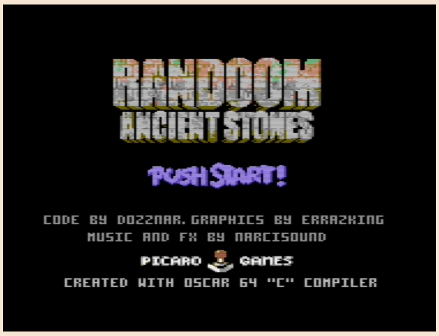
At the end of each level, there is a "black hole" that must be crossed to move on to the next level, which acts as a door.

The game uses the "classic" C64 controls, so make sure you have a good joystick in port 2 ready just in case!

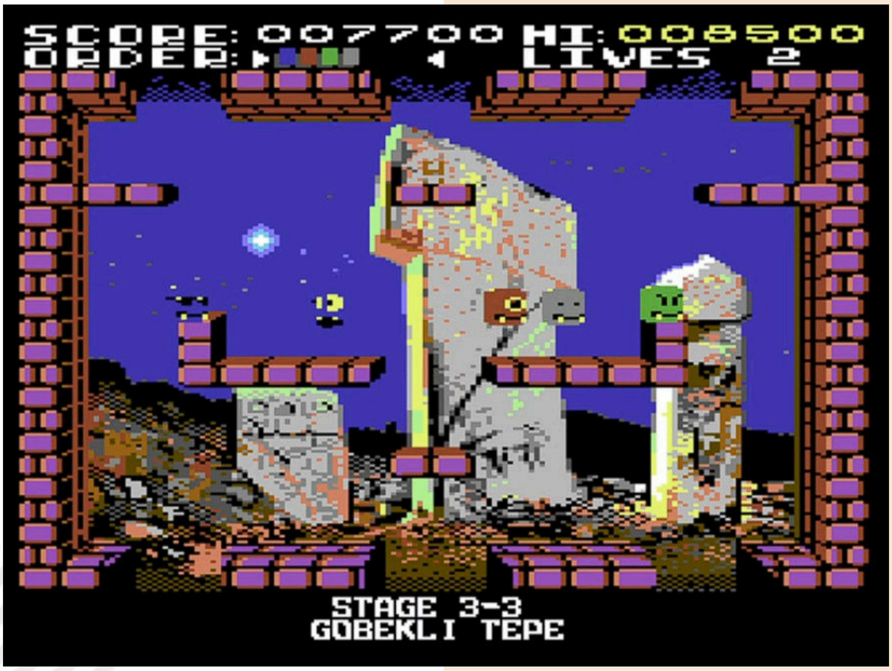
The graphics are well-designed and colorful, with really cute and squishy characters!

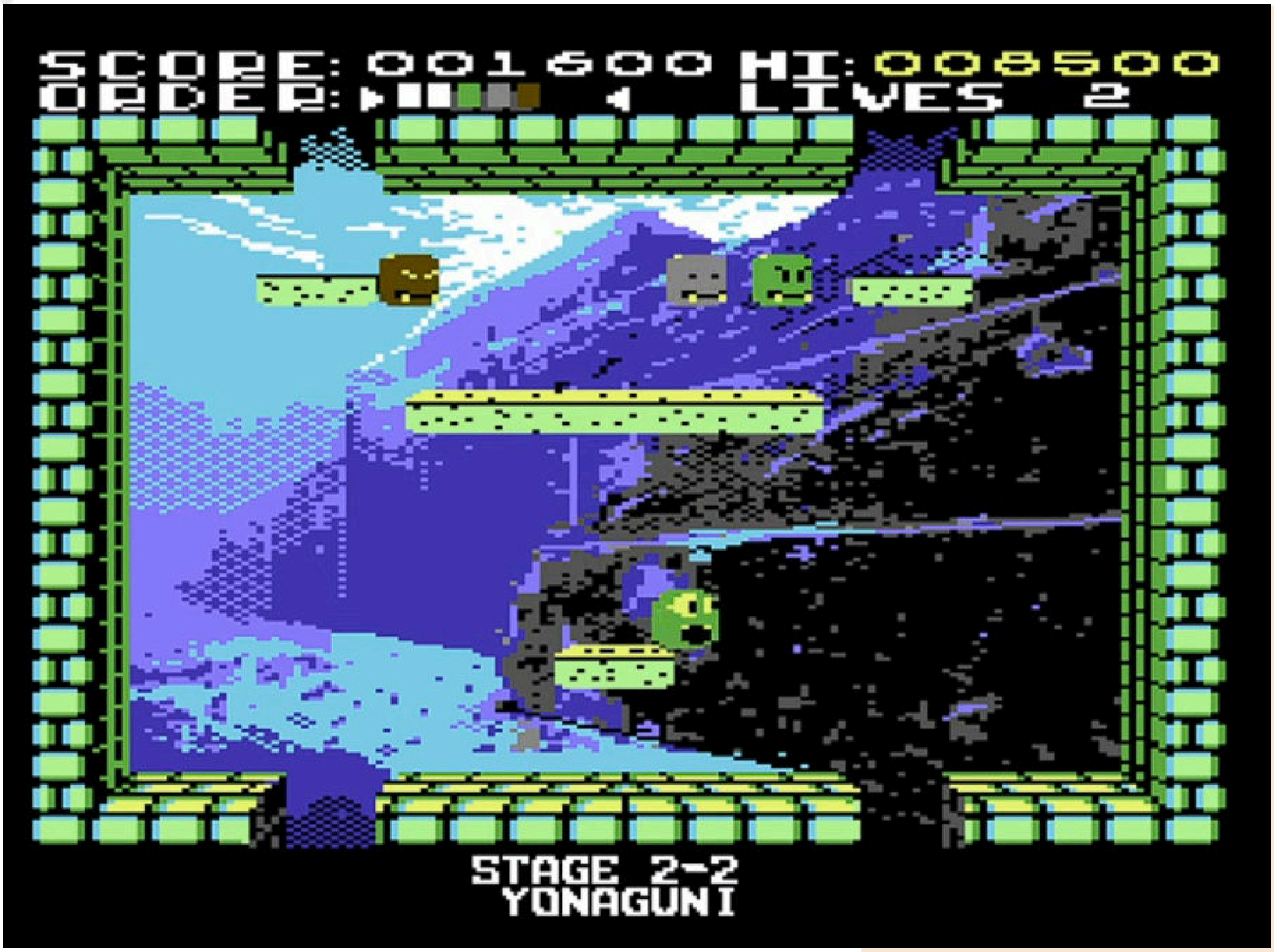


Also noteworthy is the Sid with its well-made music that will accompany you during your countless games. Because, trust Kika, this game will keep you glued to the joystick.


RANDOOM
ANCIENT STONES
PUSH START!
CODE BY DOZZNAR. GRAPHICS BY ERRAZKING
MUSIC AND FX BY NARCISOUND
PICARO GAMES
CREATED WITH OSCAR 64 "C" COMPILER

In short, anyone who loved gaming gems such as the magnificent Bubble

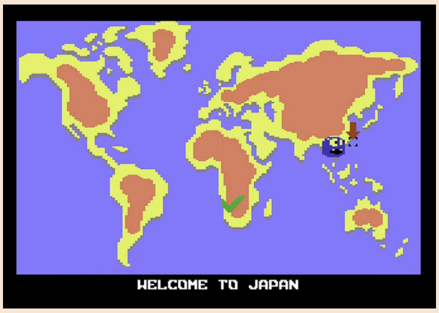


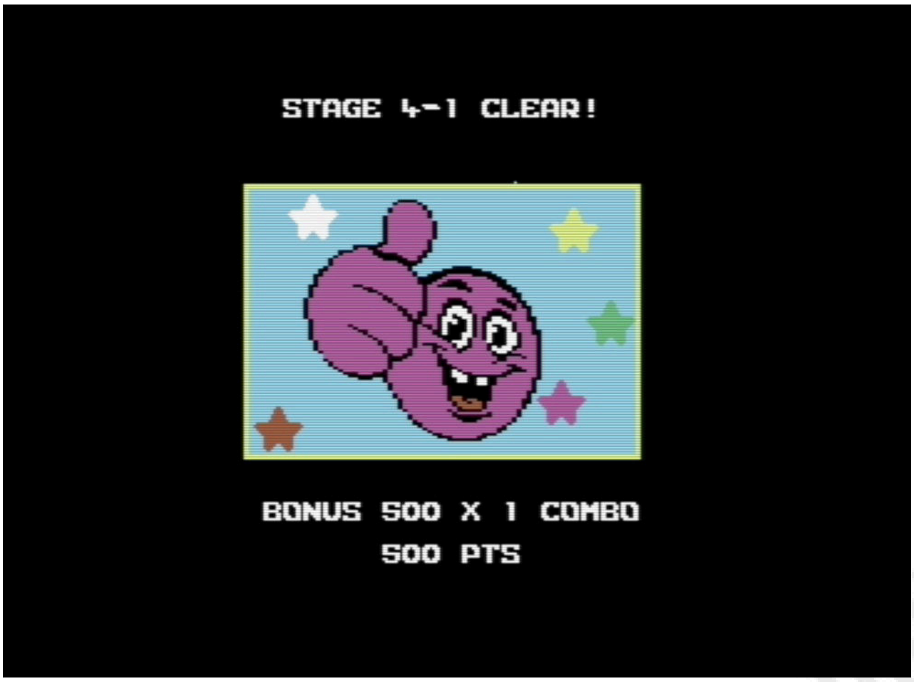


Bobble will appreciate Randoom Anciente Stone.

Finally, I would like to inform you that you can download this game on itch.io and, as always, there is also the option to leave a contribution to the developer at your discretion (but I recommend doing so, it's worth it).

by **Chiara “Kika” Carrara**





OUR FINAL SCORE

» Gameplay 90%

Interesting! I must say that it held my attention for quite some time. The controls are very simple (left, right, and jump), which makes it really playable and not frustrating at all.

» Longevity 90%

A simple game that keeps you glued to the screen. Sometimes simplicity is the winning choice. Expect lots of fun games!

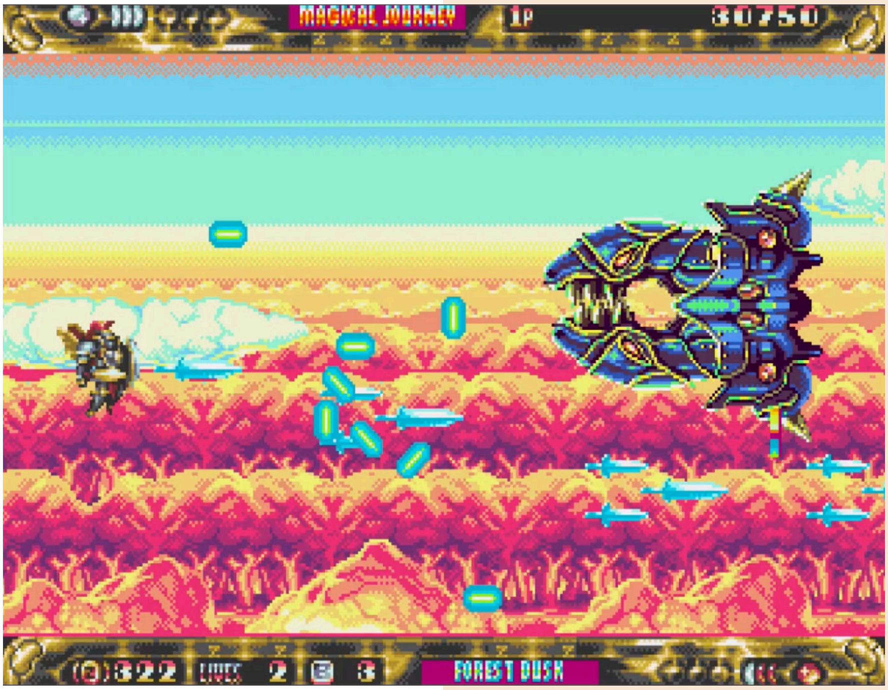
NEW GAME

ZPF

Year: 2025
Editor/Developer: Mega Cat Studios
Genre: Shoot'em up
Platform: Sega Mega Drive, STEAM
Website: https://megacatstudios.com/pages/zpf?srsltid=AfmBOoq3Q3EZPYu_Lvz7vhaqPfHWxbZOpTRGPZJkl0NZwq1Vs8BGdSiu

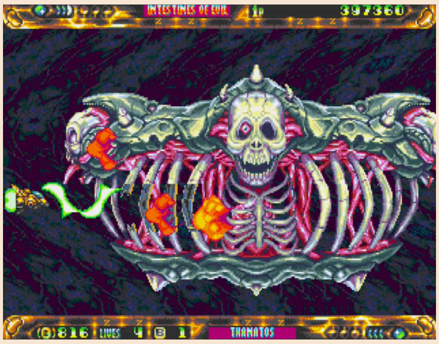
The heart of the 68000 is truly on fire! We certainly aren't short of "new" games for the Sega Megadrive, but in recent years, ZPF has captured the attention of many (players and professionals alike) compared to most other examples. After an incredibly successful crowdfunding campaign, it has finally arrived on our beloved 16-bit black console.

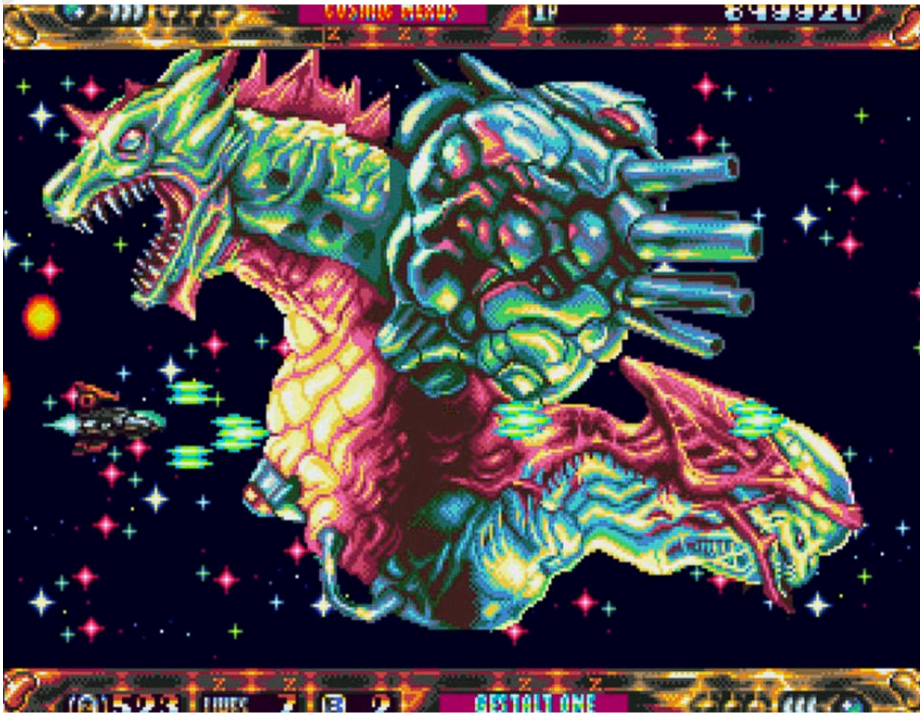
In the past, developers such as Treasure and Technosoft pushed the Mega Drive to its limits, and it is fair to say that many "modern" projects, while demonstrating incredible levels of passion, do not achieve the same result. ZPF, however, is different, as the team that developed it came very, very, very close to matching the most technically impressive releases of the 1990s with dazzling graphics, impressive boss fights, and a convincing use of parallax scrolling. Set in a bizarre and wonderful world that blends fantasy and science fiction, ZPF unfolds across seven visually distinct levels, the first three of which can be tackled in any order. Similarly, the three playable spaceships are different from each other: Gold is a spaceship with powerful concentrated projectile attacks, while Gladius has weaker shots distributed in a widespread pattern, offering greater coverage. Knight is the exception to the trio; it is not a spaceship, but literally flying armor and has the most powerful melee attacks. This feature, which requires the player to get close to the enemy, can be risky. Each spaceship also has a bomb attack.



Having three playable ships already increases replayability, but multiple endings amplify it further, and you'll also need to obtain a hidden key in each level to access the "best" possible ending. This is certainly not a game you finish once and then abandon.

There are no weapons to collect during the game, which may seem like a flaw to some players, but it actually offers a deep and nuanced system. Eliminating enemies earns you credits that can be spent between stages to upgrade your ship, add a score multiplier, get a "continue," unlock hints, and even reduce the density of enemy bullets. Losing a life slightly reduces your currency, but you can always adjust this in the options menu (as well as reducing the difficulty, increasing your lives, and the number of screen-clearing bombs you start the game with). The ending is influenced by your score, and you can





use currency to increase your points. Visually, ZPF is quite different from any other shooter from the 16-bit era. Sure, titles like Gynoug and Cho Aniki are quite unique, but ZPF's dizzying mishmash of worlds offers a truly mind-blowing experience: in one level, you infiltrate a goblin's castle, while in another, you fly over the neon-lit streets of a Blade Runner-style cityscape. Each level features an impressive boss battle, with some of the best examples literally filling the screen.

Despite the obvious cohesion issueâ€”at times it feels like you're playing three different shooters at onceâ€”the graphics are incredibly good, and had it been released during the console's original lifespan, ZPF would surely have been praised as one of the most graphically impressive shooters of its generation. The music is also excellent, making superb use of the Mega Drive's audio hardware.

Even on the lowest difficulty setting, ZPF offers a challenging experience, without ever feeling completely unfair. However, the eclectic graphics can sometimes cause problems: bullets sometimes blend into the sometimes garish backgrounds, making them difficult to spot and causing some unfair deaths. In any case, the longer you spend playing, the easier it is to "tune in" to the graphics and spot these dangers.

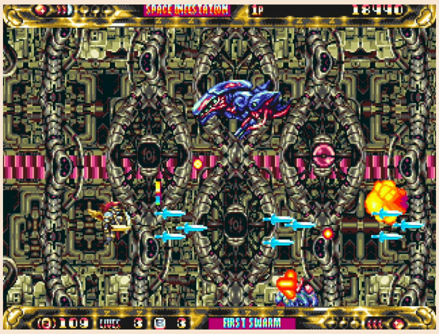
On a purely technical level, ZPF is a programming masterpiece. Few Mega Drive games achieve this standard of graphical brilliance, and the audio is equally captivating, reminiscent of some of Sega's best tunes from the 1990s. Add to this a game structure that rewards replayability and a unique upgrade system, and you have a shoot 'em up experience that can easily be ranked among the best ever seen in 16-bit gaming. The premise may be a little too risky for some players, especially traditionalists of the "spaceship versus waves of aliens" genre, but anyone who loves this genre should definitely try ZPF, either on the original hardware or on a modern console.

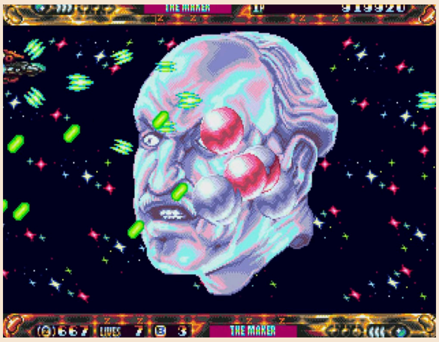
by **Carlo Nithaiah Del Mar Pirazzini**

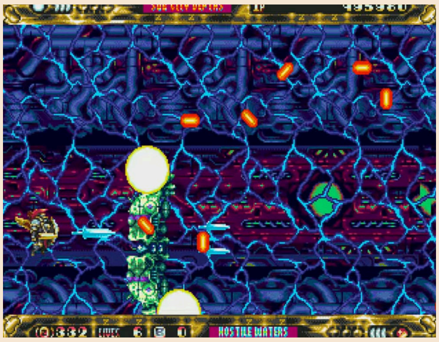
OUR FINAL SCORE

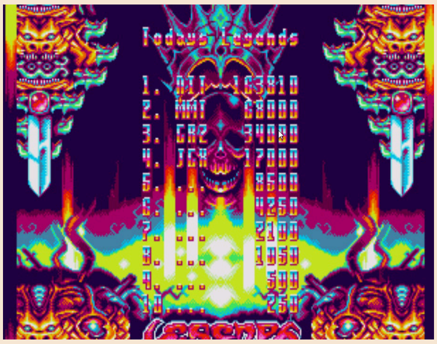
» Gameplay 90%
Good level design and interesting, atypical gameplay for this genre. The three spaceships add depth to the game.

» Longevity 90%
Quite difficult to complete... but it keeps you glued to the joypad!









NEW GAME

TIGER-HELI

Year: 2025
Editor/Developer: Atari, Plaion
Genre: Shoot'em up
Platform: Atari 7800
Website: https://atari.com/products/tiger-heli-7800?srsltid=AfmBOoo9aJ-q9Dtqr8eu34hBm-ixiyXhpebU28e_fnCwVMu01rhUI2cd&ref=retrododo.com

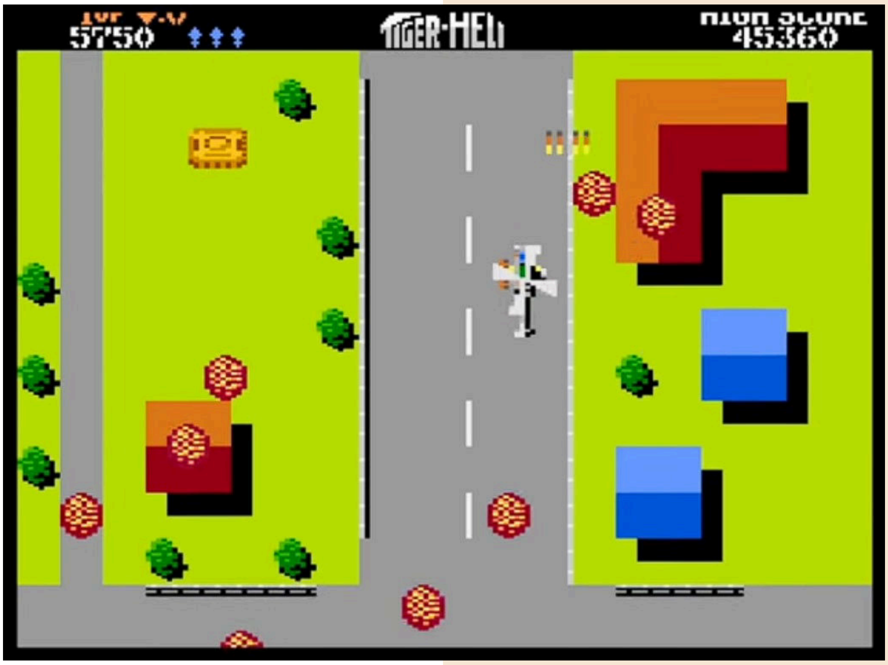
There is nothing more exciting than the return of a legend. Forty years ago, an arcade game burst onto the scene with such force that players covered their eyes in disbelief.

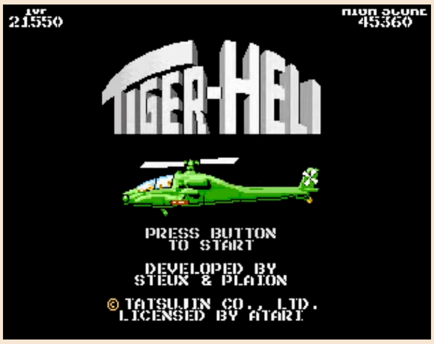
Toaplan's Tiger-Heli is rightly considered one of the most influential shooters of all time, so it's a bit surprising that it wasn't ported to more home consoles.

For many years, Micronics' much-maligned NES conversion was the only way to enjoy it outside of arcades, and although it appeared in several compilations in subsequent years, the lack of conversions in the 1980s is curiously at odds with Tiger-Heli's considerable impact on the genre.

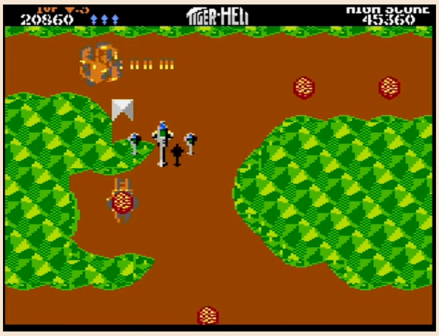
One console that could have really benefited from a conversion of the game at the time was the Atari 7800, which was struggling to replicate the commercial performance of its predecessor, the 2600. Fortunately, Atari and Plaion, with the assistance of talented programmer Bruno Steux, corrected this oversight a few decades later.

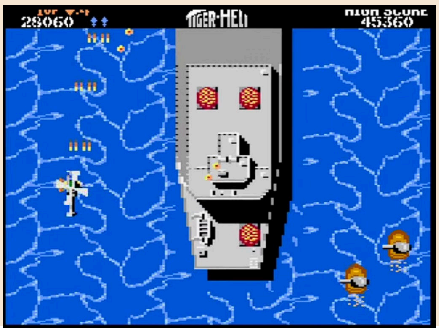
Released with the blessing of Toaplan (or rather, technically the blessing of Tatsujin, the company founded by former collaborator Masahiro Yuge, owner of the intellectual property), the Atari 7800 version of Tiger-Heli is an impressive technical feat, even if it pales in comparison to the original arcade version. In terms of gameplay and challenge, it's the best you could ask for on a console designed in 1984, the year before Tiger-Heli hit arcades. I put Tiger-Heli to the test to see how well it works on the Atari 7800 and whether this arcade classic can still hold its own 40 years later. It's time to dodge bullets, drop bombs, and evade tanks: let's take a look at my review!





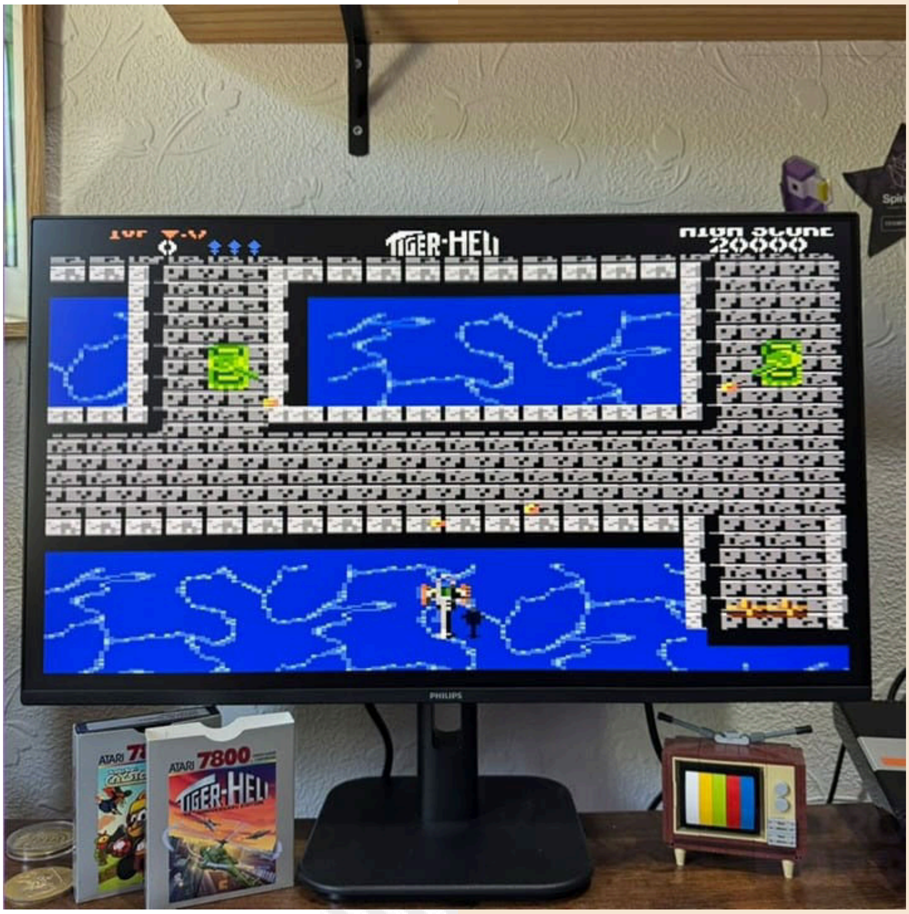
As a regular at "modern" arcades (the ones at various "nerdy" events), I continue to test my skills with old video games like PAC-MAN and Space Invaders, much more often than my bank account can afford. I love the atmosphere of arcade video games, their simplicity, the thrill of knowing that one wrong move and the game is over. It's the tension they create that makes you forget everything else in life, with the sole goal of getting to the next level.





Tiger-Heli brings all this and more to the table, or to my desk, in this case. It's a brilliant mix of arcade games and the military commando vibe that was the only thing on our minds when I was a kid in the 90s.

But let's take a look at what's inside the box developed from scratch. The box is an absolute masterpiece; in a world where everything is plastic and game cards most often don't contain games, it's great to lift the cellophane wrapper off a cardboard box and open it to find an instruction booklet and a secondary game box inside. The instruction booklet tells the whole story and the commands you learned to know and love when opening games as a child, and perfectly sets the stage for playing. The game cartridge box is convenient and fits perfectly. Both mirror the graphics of the main box, although I've always found it strange that the game graphics were on the back of the console and not visible as they were with the N64 or SNES.



But let's get to the meat of the matter! The first thing I have to mention before anything else is the soundtrack; it may be simple, but guys, it's really catchy! It's been stuck in my head for days, and I have to admit I've listened to it a lot, so it would be strange if it wasn't, but it's a fantastic tune. And even though it's repetitive and short, it never gets annoying. On the contrary, it keeps you energized and humming along, focused on the task at hand. The gameplay requires the reflexes of a cheetah and the tactical mind of Napoleon. It's fast-paced, and you have to think at least three moves ahead to avoid the numerous bullets fired at you. You take control of a Tiger-Heli helicopter (hence the name), with the mission of taking down an enemy nation determined to conquer the world. I love how the tanks seem to intuitively know where I'm going to go every single time; it makes the game interesting, and I have to time my shots perfectly if I want to get to the next level safely. Tiger-Heli consists of 4 levels, all set, as Atari rightly explains, in different environments. You'll fight by sea and land, facing 12 different enemy aircraft and vehicles, including ground vehicles, other aircraft, and turrets that will try to blow you up. You can boost your firepower with power-ups, including dropping bombs that defeat all enemies on the screen, giving you some breathing room until the next wave of attacks.

I already mentioned that the game has been remade from scratch for the Atari family of systems, and it looks great on my monitor. Let's not forget that this is a faithful reproduction of a 40-year-old arcade classic; we could never have achieved completely remastered graphics or super 3D sprites on the Atari 7800. They would have completely ruined the gaming experience; this is an anniversary edition, not a remake. That said, it will always be a bit of a change for younger players who want to get into the old Atari classics. But for older players who honed their skills in arcades and retro enthusiasts like me who spend hours and hours trying to beat their high score on The Simpsons arcade game, then this will be a nostalgic dream come true.

As I come to the end of this article, I can say that Tiger-Hei is a gaming masterpiece, comes in a beautifully designed box, and is compatible with both the original 7800 and the newer 2600+/7800+. If you're a fan of the shooters created by Toaplan in the 80s and 90s, or if you simply want to rediscover the iconic arcade shooters that brought us to where we are today, this release from PLAION REPLAI will satisfy all your needs.

As I've said in this article and countless times before, sometimes it's just nice to leave the item collecting behind, forget about solving puzzles, and just blow up enemies with simple gameplay that hits the mark over and over again. It can be a little tense at times, but it's definitely great for relieving built-up stress!

by **Alex “LEX” Fridman**

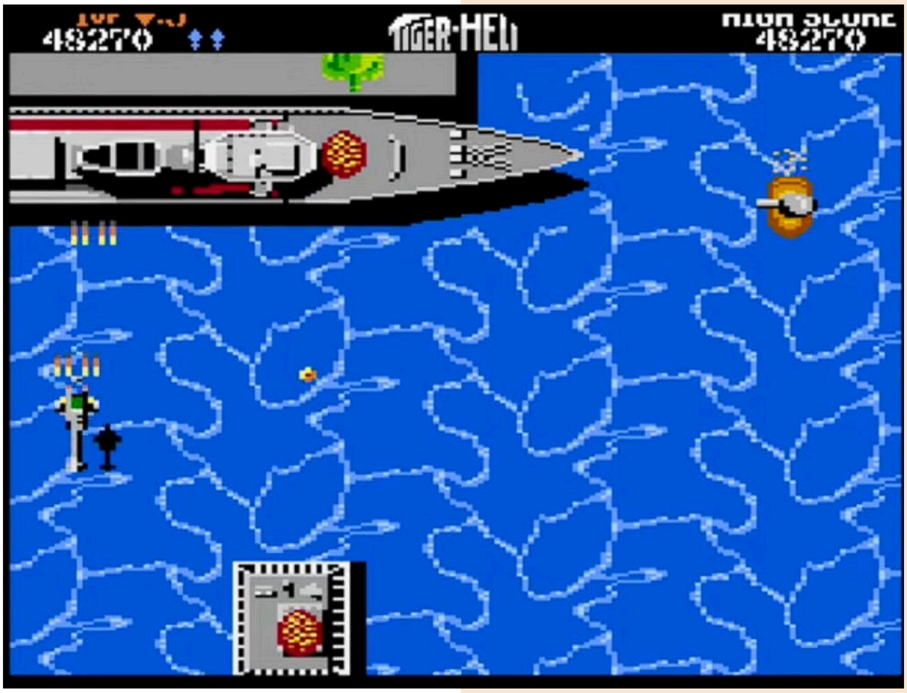
OUR FINAL SCORE

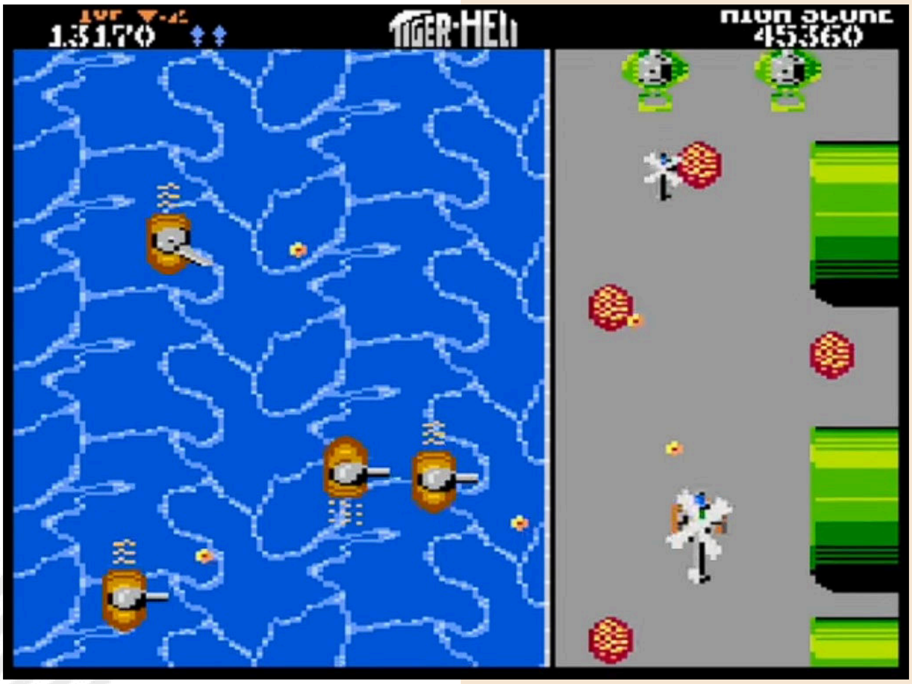
» Gameplay 92%

A product remade from scratch for Atari? Wow! An incredible soundtrack and nostalgic gameplay, topped off with fantastic packaging and an affordable price.

» Longevity 89%

It's not an easy game for all the reasons I listed in the review. It's a classic, and as such, it's hell in the hands of a young adult gamer.





NEW GAME

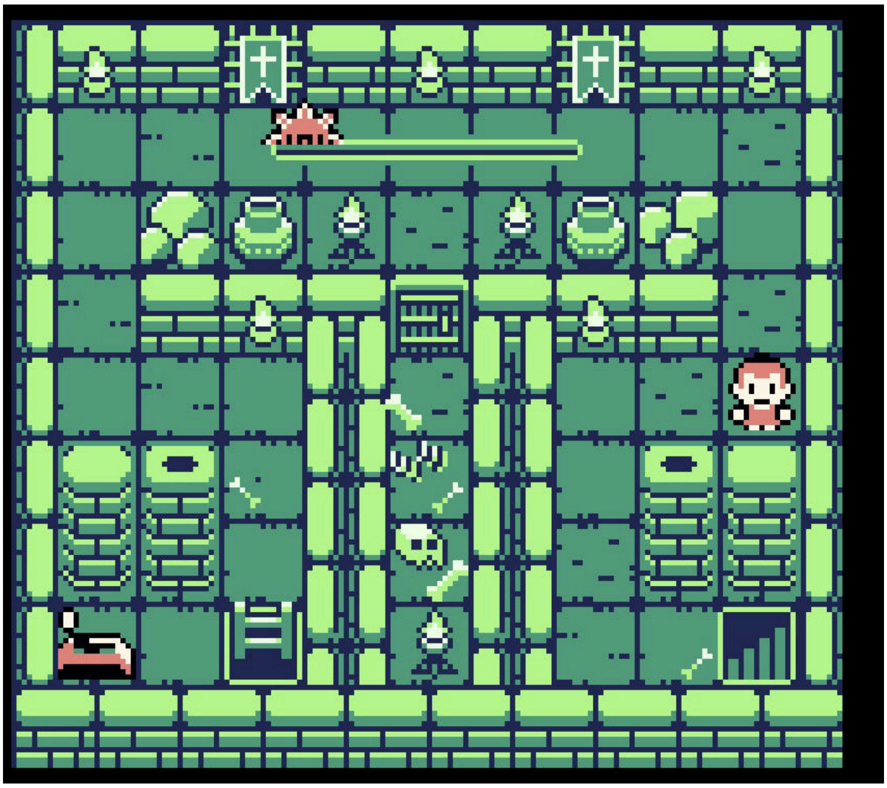
CASTLE OF TERROR

Year: 2025

Editor/Developer: PCNONOGames

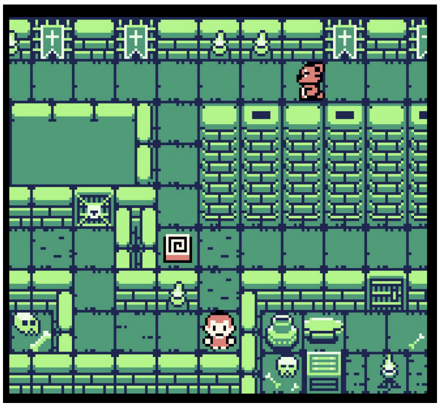
Genre: Puzzle

Platform: Game Boy Color



Castle Of Terror is super fun! The gameplay mechanics are incredibly simple, making it accessible to everyone.

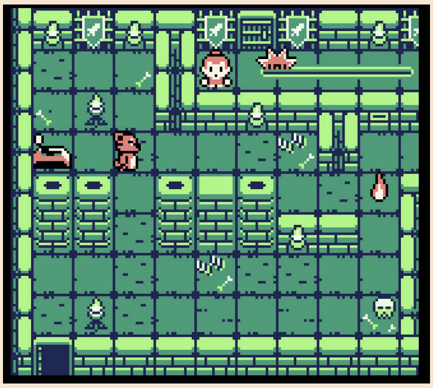
The pace of the game is great, and levels can be completed in a matter of seconds. It's an accessible but challenging game. As you progress, the difficulty curve increases.
The different enemies, obstacles, and objects add variety to the game. Furthermore, this variety is integrated with a well-balanced approach that ensures the player never loses interest.

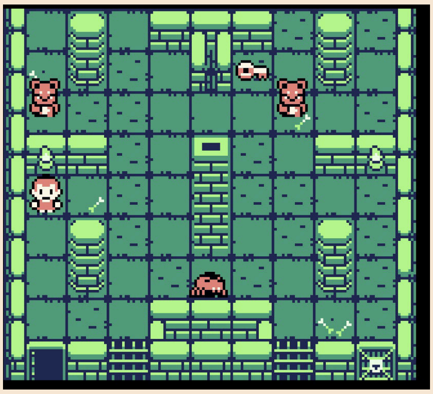


The music is fantastic. It fits the game perfectly. It is not tiring, distracting, or annoying, but perfectly accompanies the stages of the game. The compositions are tasteful, and the author's experience in both creating music for video games and designing audio chips for 8-bit machines is evident.

In short, a fantastic and super fun game, clearly made with great care and skill.

by **Carlo Nithaiah Del Mar Pirazzini**





OUR FINAL SCORE

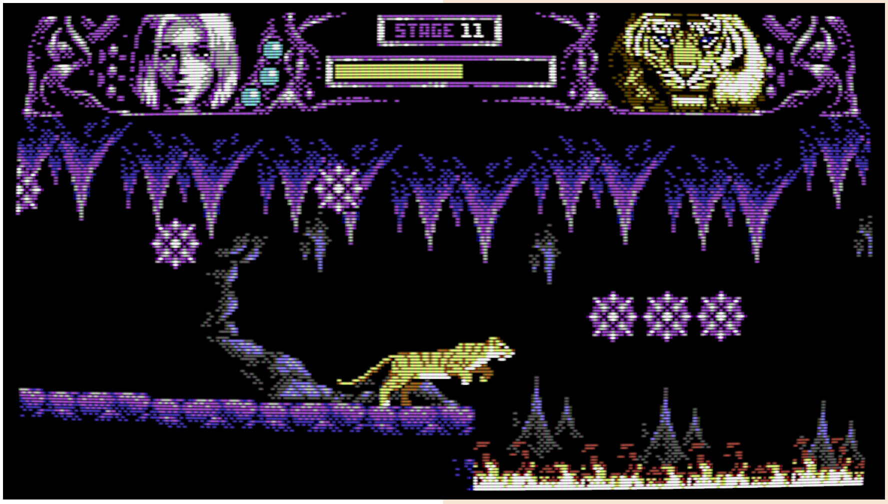
» Gameplay 90%
Super fun!

» Longevity 85%
It's an accessible but challenging game. As you progress, the difficulty curve increases!

NEW GAME

JIXA
LADY TIGER

Year: 2025
Editor/Developer: Oscar Celestini
Genre: Action
Platform: Windows
Website: https://oscarcelestini.itch.io/jixa-lady-tiger



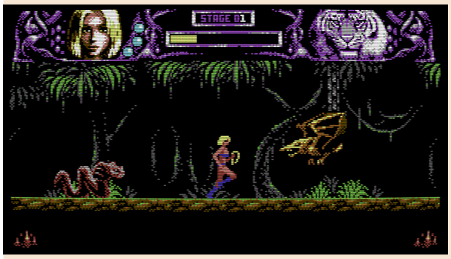
Jixa Lady Tiger strongly evokes the atmosphere of the C64. Oscar Celestini, the developer, has given the gaming world a title that pays homage to the great C64 games.

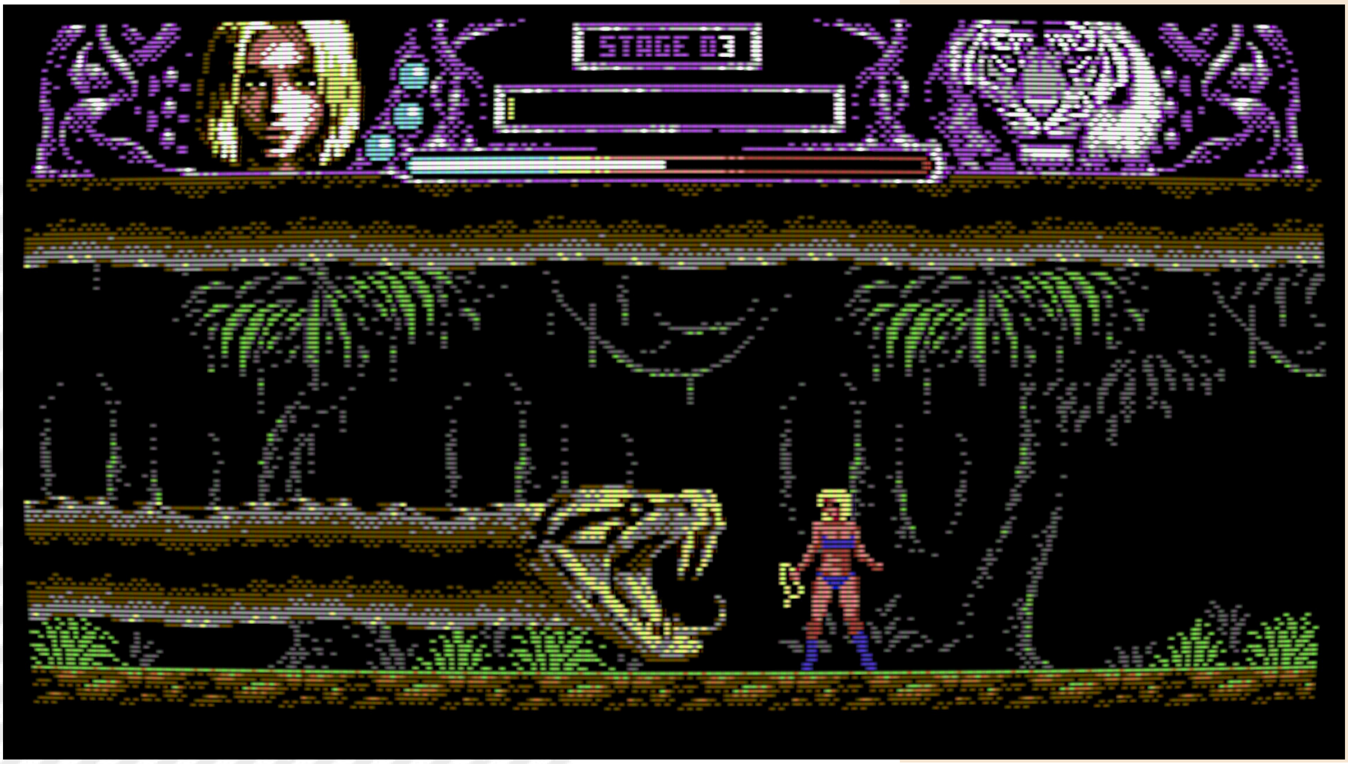
The atmosphere is reminiscent of Intelligent Design's 1988 hit Vixen. Jixa resembles that great game, almost becoming its heir in the new era of retrogaming.

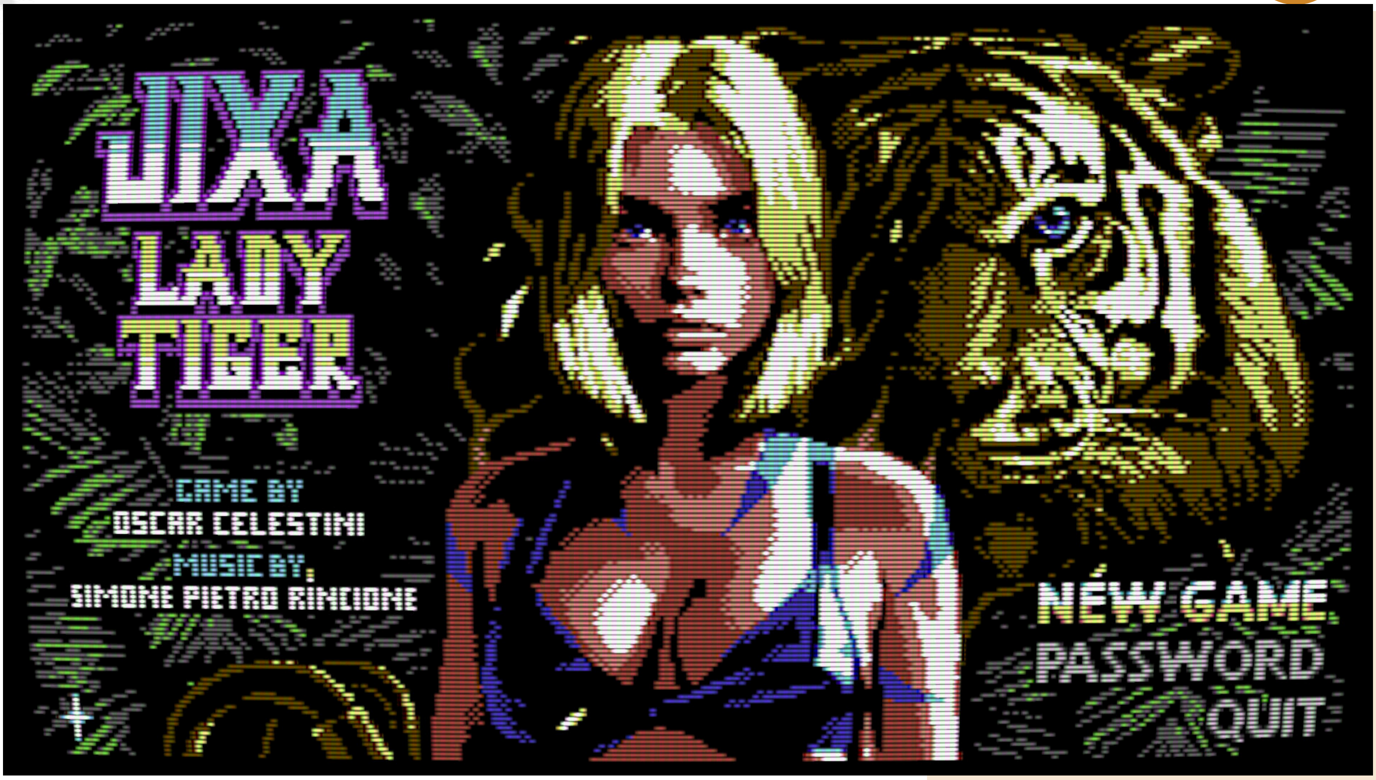
It is a nostalgic title that Celestini created using the C64 color palette. We play as a warrior armed with a whip who can transform into a ferocious tiger. This is an evolution of the fox transformation seen in Vixen.

Our heroine will have to overcome a handful of levels, some in human form, others as a tiger, and defeat monsters and fearsome final bosses. The transformation only takes place by collecting certain mystical icons present in the level.

It's an "old school" style title, fast-paced, pure arcade fun that will appeal to those who have experienced this







kind of game in the past. The developer has even included a CRT video filter to enhance the nostalgia effect.

Technically, it is pleasing and beautifully animated, with a really well-made soundtrack written by Simone Pietro Rincione.

It costs only 5 euros on itch.io but is well worth it. Be warned, though! It's a difficult game that requires nerves of steel, patience, and a lot of willpower.

Not recommended for beginners or those looking for something more relaxing.

by **Giampaolo Moraschi**

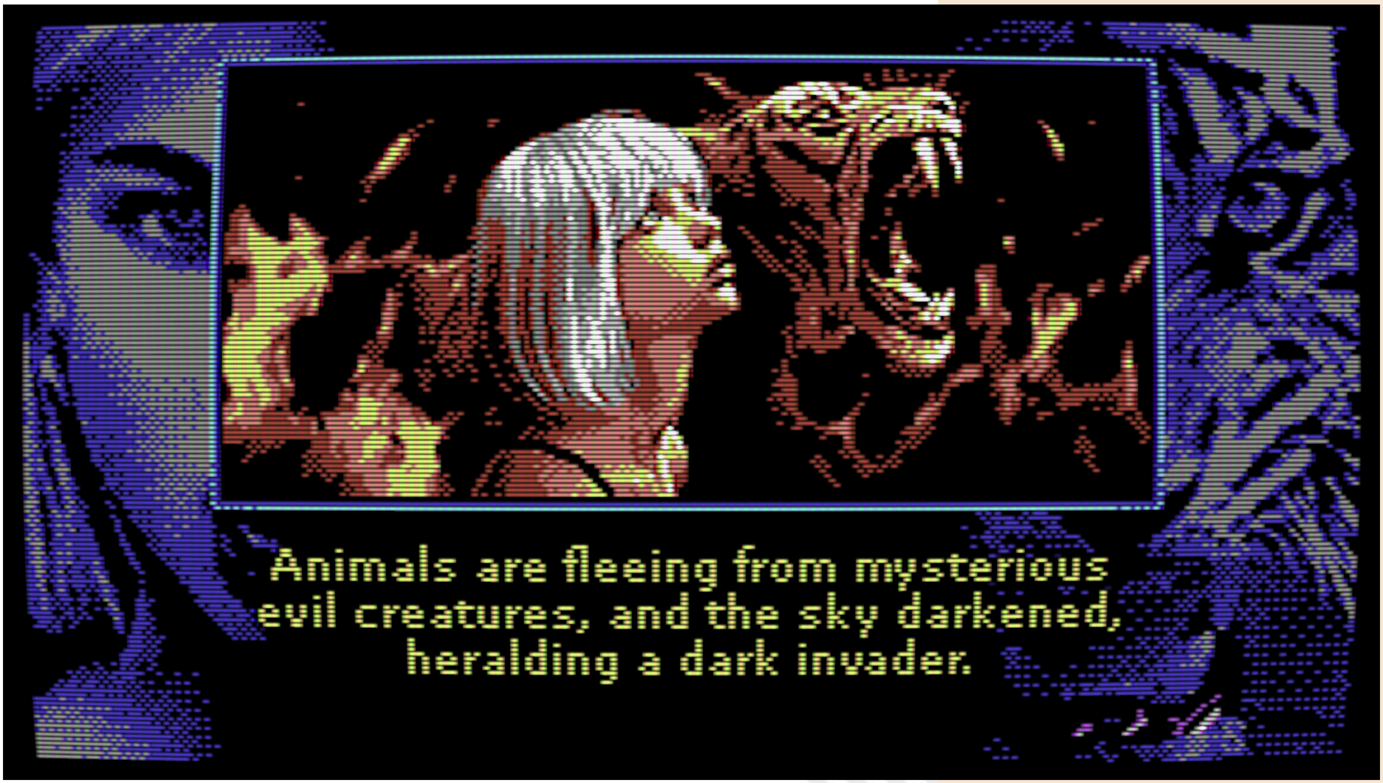
OUR FINAL SCORE

» Gameplay 85%

There is a feeling that links the 80s to modern titles. A dynamic game with great attention to detail.

» Longevity 80%

Ruthless but great fun!



Animals are fleeing from mysterious evil creatures, and the sky darkened, heralding a dark invader.

NEW GAME

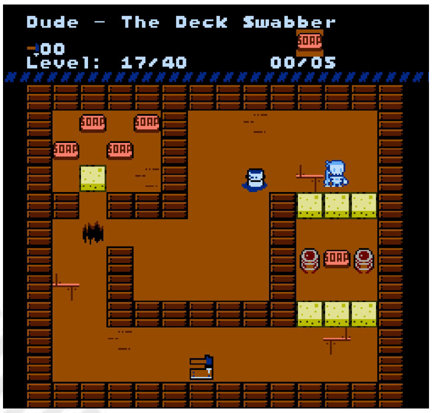
DUDE THE DECK SWABBER

Year: 2024
Editor/Developer: Richard Oz
Genre: Puzzle game
Platform: Nintendo NES
Website: https://richard-oz.itch.io/dude-the-deck-swabber

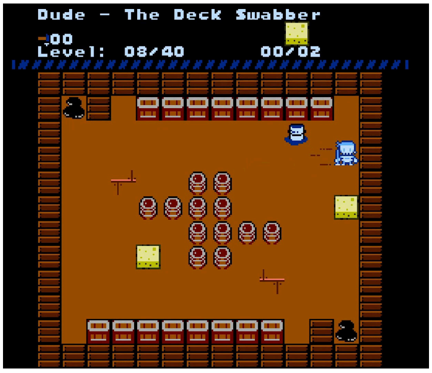
"Join the pirates, he said! Yeah, right! You'll see the world, he said! All I see is my broom, a bucket, and the filth on this ship."

Dude is a deck swabber. He's a few centuries old and looks more like a skeleton than a young sailor, but he doesn't care. He joined the ship to become a real pirate... even if the road ahead is definitely uphill.

In this adventure, he must clean 40 rooms on the ship by solving several puzzles. Not everything is easy. He'll have to move boxes, collect sponges, and sweep everything thoroughly. Once he's cleaned up, he'll move on to the next level, increasing the difficulty of the challenge.



The game was developed with NESmaker and is a rather well-made and entertaining work. By downloading the ROM file, we can use it in emulation or perhaps flash it to a cartridge or put it in Everdrive. The file also includes a nice STL model to print in 3D, a much-appreciated gadget.

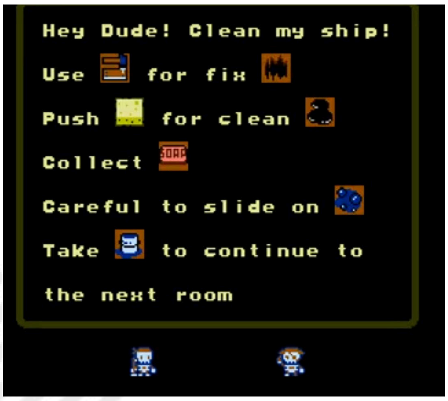


Getting to the heart of the game, Dude is a fun title with the classic style of fixed-screen puzzle games. The puzzles are simple but well constructed, and there are never any hiccups in the level. You can also "repeat" wrong moves by pressing the B button on the pad.

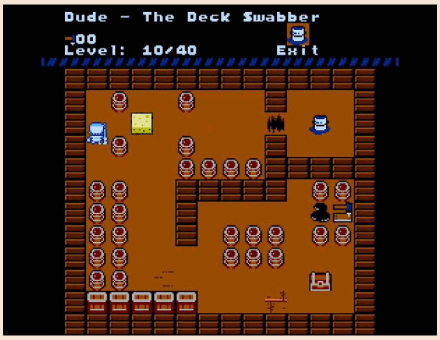
Visually appealing and with simple sound effects, Dude is a game you can play between more challenging titles, perhaps during breaks or with friends/children/spouse.

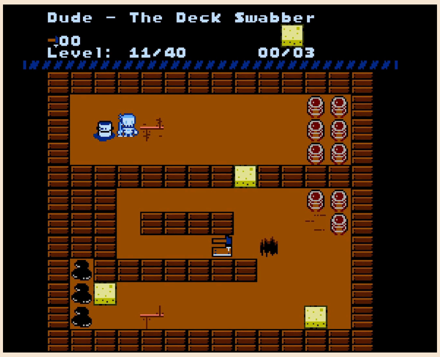
In short, it's cute and cuddly!

by **Marta Rossmann**



Hey Dude! Clean my ship!
Use for fix
Push for clean
Collect SOAP
Careful to slide on
Take to continue to
the next room





OUR FINAL SCORE

» Gameplay 80%

Few controls and all simple.
The levels are well done.

» Longevity 80%

A good 40 levels of riddles and puzzles will keep you busy for quite a while.

NEW GAME

DEFENDER

Year: 2025
Editor/Developer: Sark02
Genre: Shoot'em up
Platform: Atari ST
Website: www.github.com/therealsark02/defender

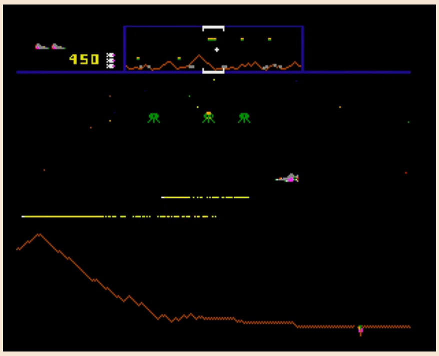
The Atari ST boasts a small but brilliant selection of Defender-style games, including excellent alternatives such as Anarchy and StarRay. Of course, there is also the classic Defender II by ARC, one of my favorites with its style and elegance. However, I don't remember ever playing an official version of Williams' Defender.

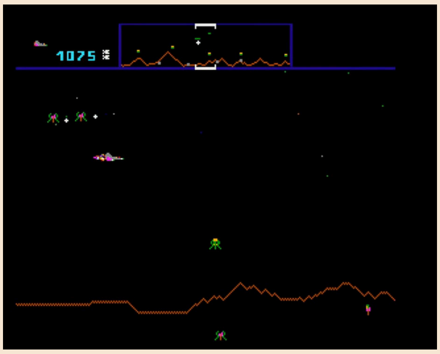
This year, developer Sark02 started and completed an arcade-perfect version of Defender. This involved reverse engineering the game functions by translating the 6809 code and modifying it to work on the ST/e. The resolution of the original is similar to 320x256, but instead of losing pixels, he implemented overscan to remove the bottom border. Not only that, but it also uses DMA audio hardware for 25 kHz playback, while supporting the STf.

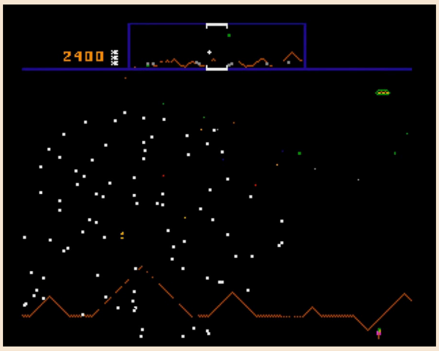
Now, unless you've been hiding under a rock for more than 40 years, you'll know what Defender is. A fast and intense shooter that marked an era. No one was as frantic and fast as it was in the arcade. Aliens to kill and humans to save before they were hybridized. Frantic, full of spectacular explosions, and tough as nails. A true arcade classic.

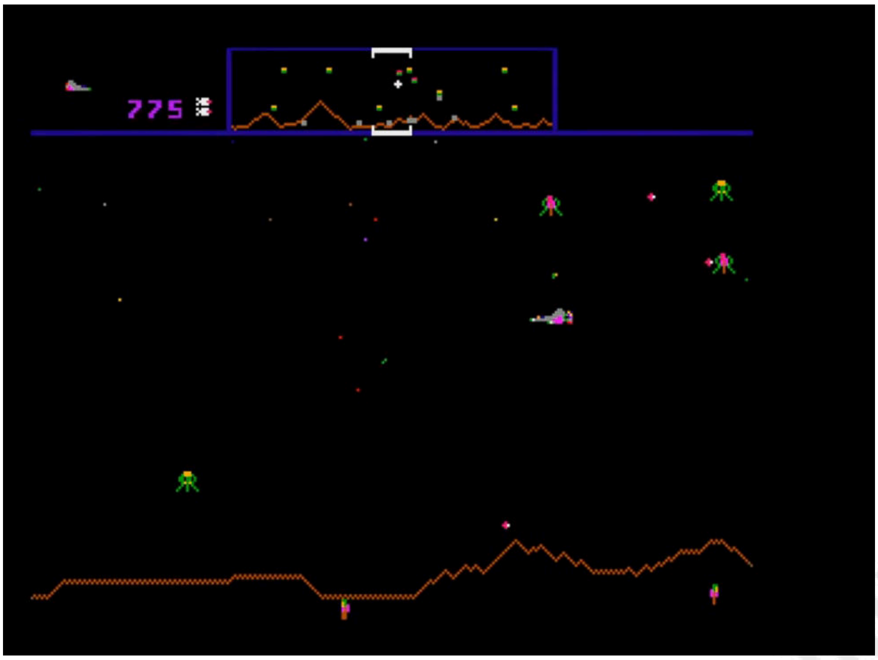
This ST version is truly "arcade perfect." A masterful conversion for the ST (but even better on the ST/e). Rebuilt around the original code and enhanced with overscan graphics, smooth controls, and crisp DMA sounds. Sark02's commitment shines through in every pixel. Incredible!

by **Giampaolo Moraschi**









OUR FINAL SCORE

» Gameplay 95%
Perfect reproduction of that beautiful and highly playable arcade masterpiece.

» Longevity 95%
Tough as steel but as enjoyable as Coca-Cola in summer.

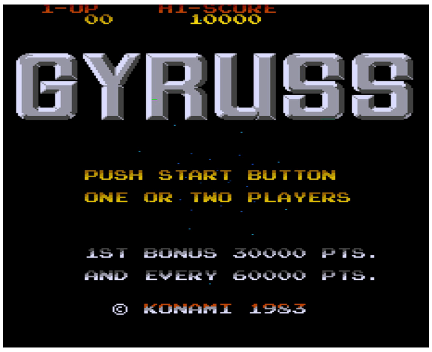
NEW GAME

GYRUSS

Year: 2025
Editor/Developer: JOTD
Genre: Shoot'em up
Platform: Amiga 1 MB di ram
Website: https://jotd666.itch.io/gyruss

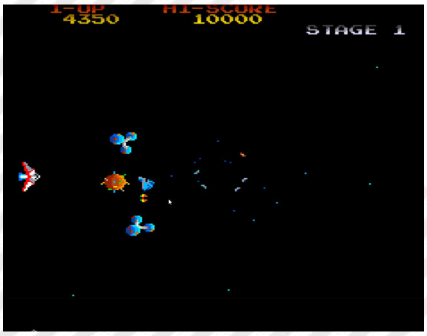
Gyruss is one of those arcade games that would grab you and make you spend all your weekly allowance.

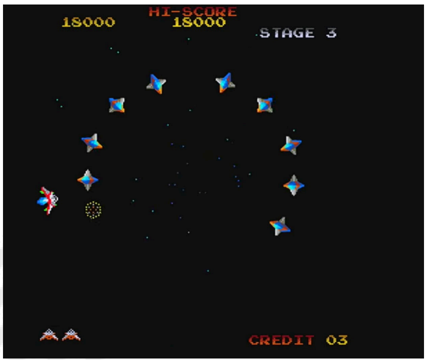
It was a shooter that changed the way we played games. No more aliens from above! Make way for three-dimensional (fake but impressive) waves that rotated across the screen, creating an immersive gaming experience.



Yoshiki Okamoto and Konami changed the way we looked at this genre of games, which was already rather overused back in 1983.

Most of the enemies are spaceships, which must all be eliminated to complete the level. Enemies appear from the center or edges of the screen and attack in circular trajectories. The depth effect is guaranteed: enemies and projectiles enlarge as they approach the center of the screen. The game also features other types of opponents: satellites, asteroids, and laser cannons. These appear intermittently and move away after a few seconds.





The title was a huge success in arcades and had some great conversions for home systems. Among these, the C64 version stood out as very well done, as did the NES version, which also had an end-of-level boss.

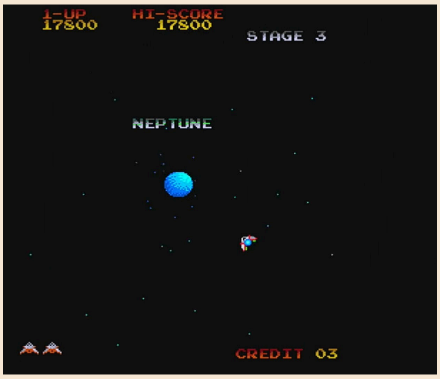
Nothing on Amiga until now, with the excellent version created by JOTD, now an expert in game reverse engineering from M6809/Z80 to 68000.

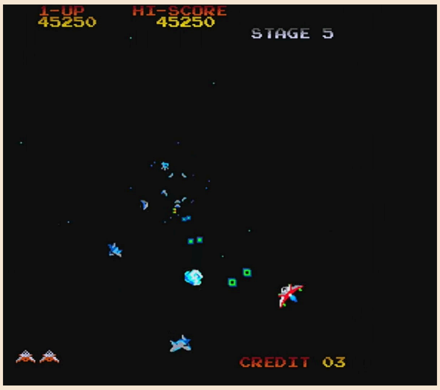
The game is complete, including Bach's "Toccata and Fugue in D Minor" in a rock version rearranged by Sky.

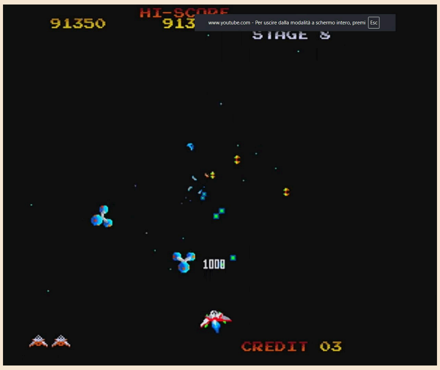
No excessive configuration is required, but we strongly recommend a bit of fast memory to better enjoy the game.

What's that? Is it still that evil and devastating, extremely difficult, coin-eating title of yesteryear? Yes, of course, and that's precisely why it's so great, because it's complex to tackle.

by **Carlo Nithaiah Del Mar Pirazzini**







OUR FINAL SCORE

» Gameplay 80%

Its rotation system is still very valid and usable. Aesthetically, it is not varied, but it is still a great game to play.

» Longevity 80%

It's a title of its era: beautifully difficult and ruthless. But we love it for that!

NEW GAME

DONKEY KONG

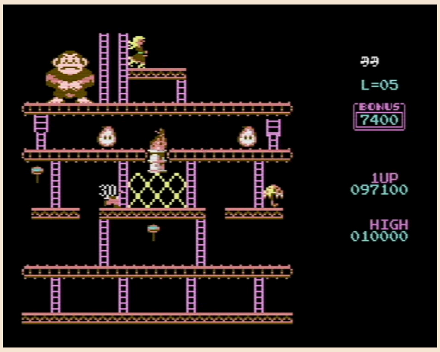
Year: 2025
Editor/Developer: TCFS, Unreal, Csabo, Chronos, Siz.
Genre: Platform
Platform: Commodore PLUS/4
Website: https://tinyurl.com/38k7z6k6

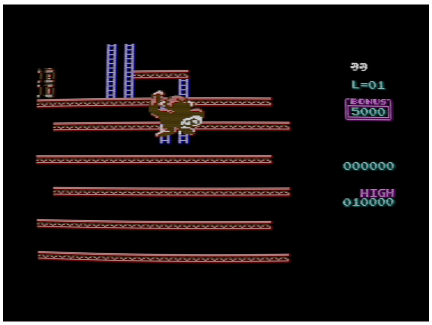
2025 is the year of Donkey Kong! On Nintendo Switch 2, we are enjoying the incredible Donkey Kong Bananza. We had a nice ZX Spectrum version and one in UGBASIC for PC128, and now it has also arrived on Commodore Plus/4 with a direct conversion of the title seen in 2016 for Commodore 64. Here too, there is little to say about the plot... there is always Pauline to save, barrels to jump over, and the big monkey to calm down. In pure Nintendo style, as it has been since 1981.

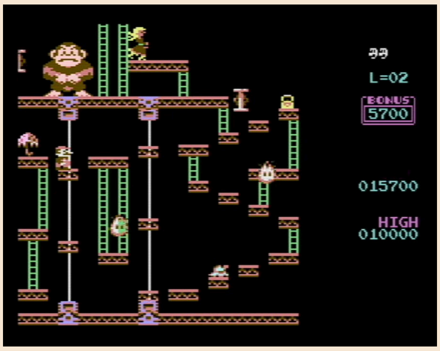
This version is practically identical to the one that came out on the C64. The graphics are beautiful and functional, the sound is there and it's just right. In short, it has everything. Almost everything!

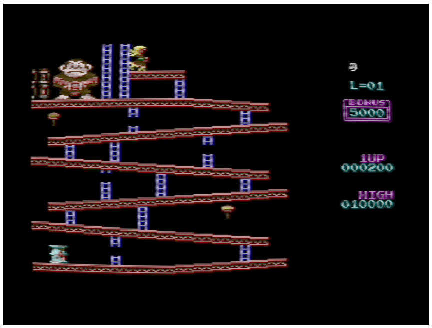
The controls are not at all precise. The jump is not well designed and you will find yourself dying for a "handful" of pixels. And in this game, precision is necessary to proceed. It's a shame because everything else is fine. For now, only one "fixed" version has been released, but little has changed. Let's hope for a revised version soon!

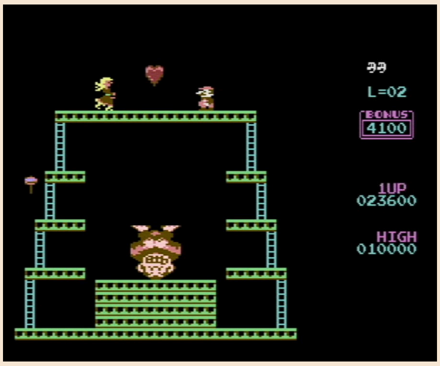
by **Giampaolo Moraschi**

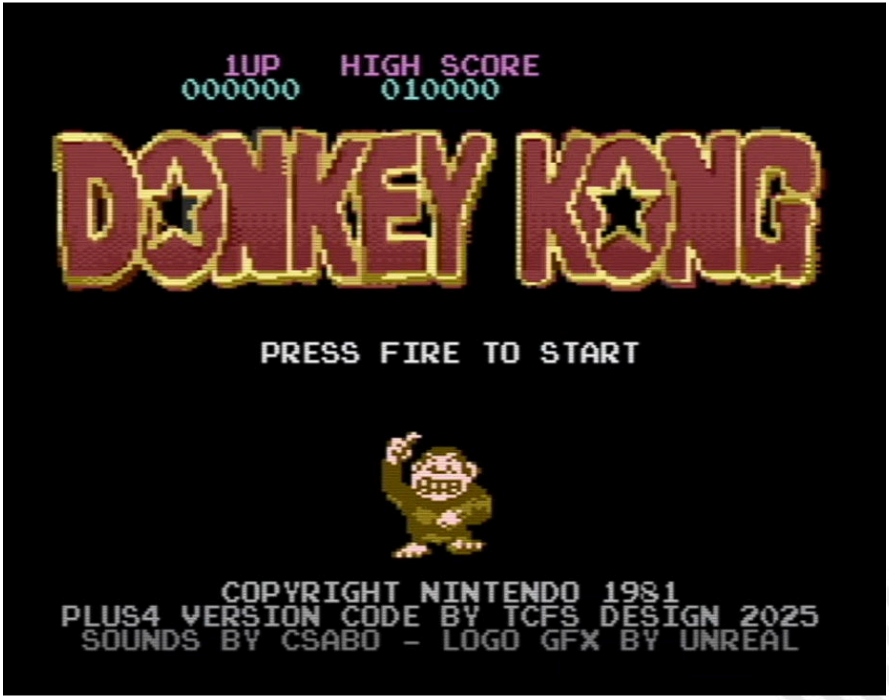











1UP HIGH SCORE
000000 010000
DONKEY KONG
PRESS FIRE TO START
COPYRIGHT NINTENDO 1981
PLUS4 VERSION CODE BY TCFS DESIGN 2025
SOUNDS BY CSABO - LOGO GFX BY UNREAL

OUR FINAL SCORE

» Gameplay 70%
It's Donkey Kong with game control issues.

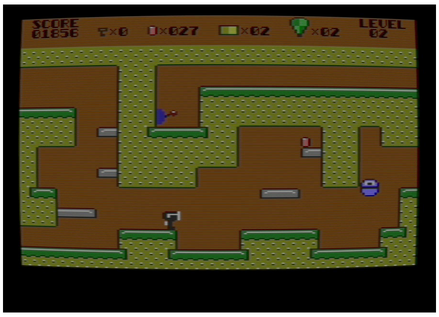
» Longevity 70%
..This problem makes you lose the desire to play.

NEW GAME

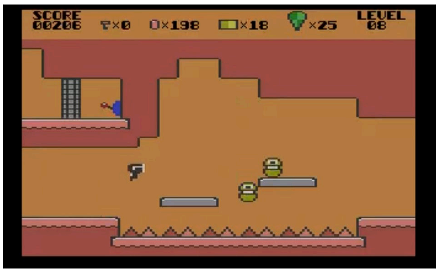
PHIL’S ADVENTURE

Year: 2025
Editor/Developer: Epy, Csabo
Genre: Platform
Platform: Plus/4
Website: https://plus4world.powweb.com/software/Phils_Adventure

Beautiful and fun! This interesting platform game for Commodore Plus 4 has just been released and, unexpectedly, has stolen quite a bit of my time.

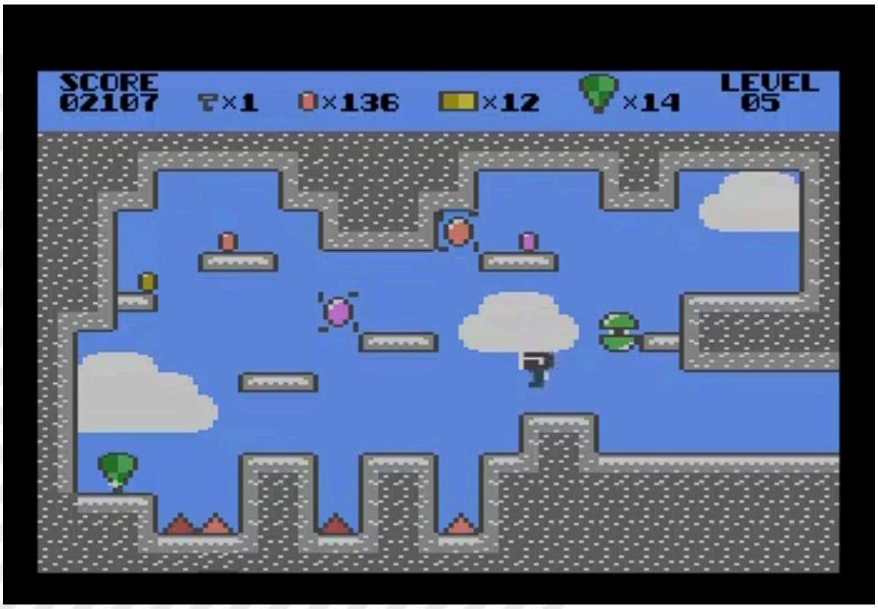


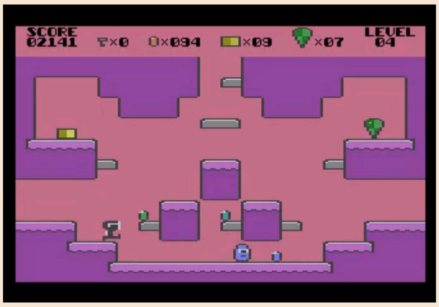
That's right, guys! Phil's Adventure has managed to grab my attention, a little game that, in just 64k, has literally glued me to the joystick.
Our goal is to help our friend Phil get through numerous levels, avoid dangerous enemies, and avoid traps. All this with the aim of activating switches and finally getting to a key that will take him to the next level.
It's an "old school" style game, and I'm sure it will catch your attention with its simple but well-designed graphics in bright colors and the protagonist's sprite that moves smoothly and precisely. It's a rewarding and relaxing gaming experience, thanks in part to the gameplay that is never too stressful.

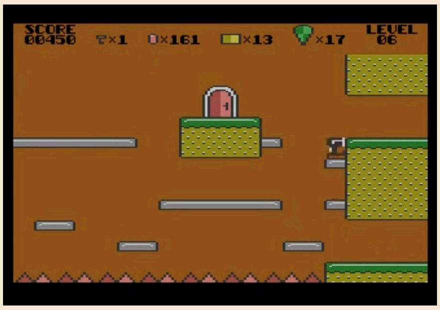


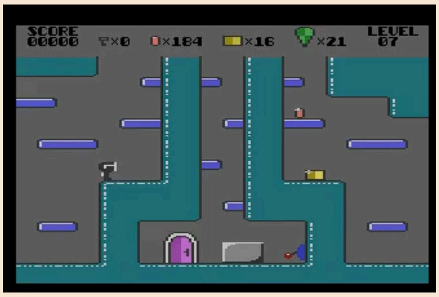
The game was developed by Epy and Csabo (for the music), a team that is now well-established on Plus/4.

by **Chiara “Kika” Carrara**









OUR FINAL SCORE

» Gameplay 93%

A platform game that doesn't innovate the genre, but still surprises with its playability. I'm convinced that you'll spend a lot of time playing it, even if at first glance it seems like just another game.

» Longevity 93%

Definitely a gem released for the Commodore Plus 4! And as I said, it will "steal" a lot of your time. What are you waiting for? Run and try it, you won't be disappointed!

WILLOW

Year: 1989

Editor/Developer: Capcom

Genre: Platform

Platform: Arcade

In 1988, Willow, a fantasy film produced by George Lucas, starring Val Kilmer and directed by Ron Howard, was released in theaters.

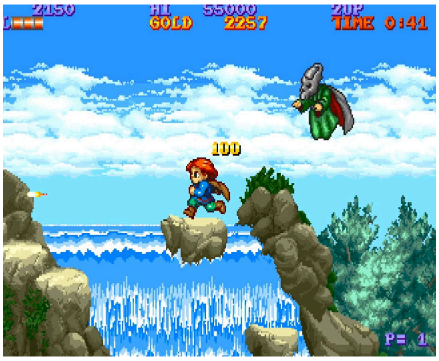
The film was a success, and so, the following year, Capcom created a video game adaptation as an arcade title.

The game is a classic platformer but with some RPG elements and immediately impressed fans with its fidelity to the film.

The protagonists are Willow Ufgoog, small and weak but with the ability to use his magic, and Madmartigan, who is a more aggressive character who attacks enemies with his sword.

But let's not be fooled, we can neither choose which hero to play nor play them together in cooperative mode, because the title itself forces us to use a specific character in the various scenarios.

Speaking of scenarios, the various levels transport the player through the film's iconic locations, such as the Nelwyn village, the Nockmar forest, and Tir Asleen castle. At the end of each scenario, we almost always find the classic boss to defeat in order to move on to the next area.



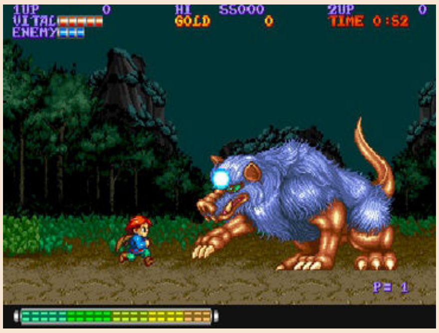
The gameplay provides us with two buttons, one for jumping and one for shooting, but by holding down the latter for a few seconds, we can charge an energy bar that will allow us to unleash a more powerful attack.

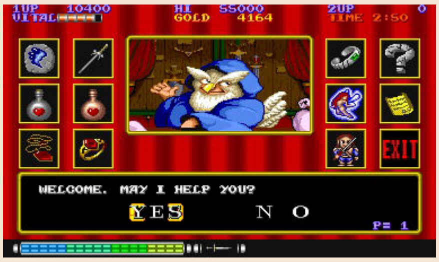
During our adventure, we can also collect gold coins to spend in a shop to buy extra lives, protection from enemies, or upgrades for our attack. The graphics are really nice, with colorful and well-made sprites, beautiful animations, and backgrounds that still recall the locations from the film.

The soundtrack is also impressive and, although not inspired by the main theme of the film, features melodies that immediately stick in your mind and sound effects suited to every moment of the game.

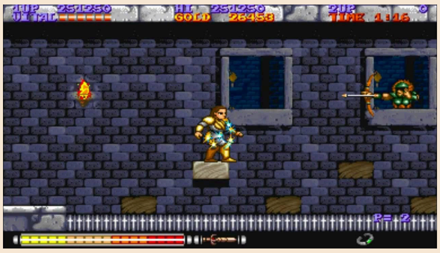
In addition to the arcade title, video games inspired by Willow were created for the NES and computers, but both have completely different gameplay from the title created by Capcom.

by **Querino Ialongo**





WELCOME. MAY I HELP YOU?
YES NO



OUR FINAL SCORE

» Gameplay 95%

Willow is a really fun platformer to play with some sprinklings of RPG elements that immediately bring to mind classic titles such as Black Tiger and Wonderboy in Monster Land.

» Longevity 90%

A detailed map shows us the six levels to tackle, while the stunning animations really give us the feeling of experiencing the various scenes from the film.

CONAN

Year: 1984

Editor/Developer: Datasoft

Genre: Platform

Platform: Commodore 64

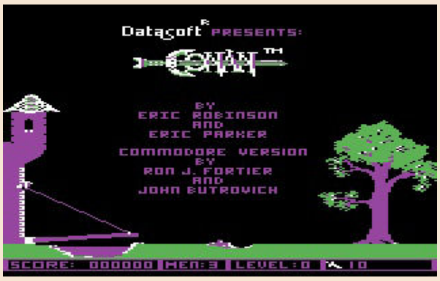
After returning from vacation and with autumn already in the air with the first cold spells and rains, I saw something great produced by one of our passionate retro enthusiasts who continues to create titles and headlines despite the fact that Commodore closed its doors more than 35 years ago. The game in question is Musketeer, a brand new and eagerly awaited title that will certainly be reviewed on these pages. Musketeer couldn't help but remind me of titles such as Bruce Lee, Zorro, and The Goonies, given that the gameplay is very similar... But another title almost identical to the three mentioned was about to slip my mind, and I also realized that it has been underrated.

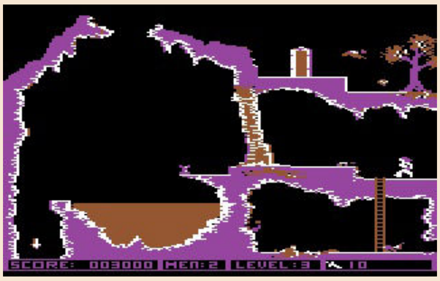
It is Conan, a video game based on the 1980s film of the same name starring Arnold Schwarzenegger. No, I am not talking about the perhaps better-known text adventure, but rather the Bruce Lee-style platform game that, as mentioned, went somewhat unnoticed. The game includes seven levels of increasing difficulty, and in each one, we have to find the exit, but not before performing certain actions using the objects and furnishings in the levels to unlock it, such as freeing prisoners with the appropriate key that must be found and grabbed.

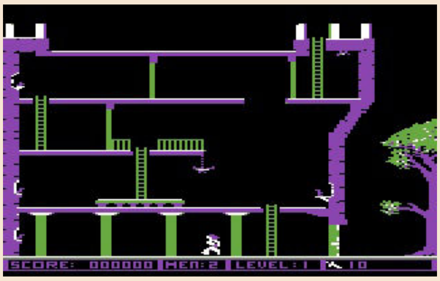
The first level is very simple: you just have to avoid the bat that flutters around, observing its trajectory and then moving while it is perched. Graphically, there is not much to say: the style is exactly the same as in Bruce Lee, and the sound changes at each level with nice little tunes. The sound effects are also good, although they are a little quiet, perhaps due to the overall lack of fluidity. The game may be affected by a few bugs, particularly in the imprecise and tricky jumps. This is simply because the more sprites there are on the screen, the more the game may slow down, affecting our actions such as jumps and weapon launch timing... You got it right: the protagonist is equipped with a limited number of throwing axes, but they can be recovered in the various levels. I recommend not using them excessively because they may be needed later on for important things, including solving puzzles.

However, it is a very satisfying game, and the fixed-screen levels are all different. They range from castle dungeons to underground caves. We only have two lives to complete all seven levels of increasing difficulty, but don't worry, once you get the hang of it and understand how each level works, you'll get through it without too much trouble, provided that the fluidity remains good, which guarantees good longevity and perhaps even a great desire to replay this and similar games, in addition to the aforementioned Musketeer, which has just arrived. Many of you will still be on vacation enjoying the last days of warm sunshine, evenings spent in bars and clubs with beautiful women before returning to work, which is really sad in a way. As you return, think about what awaits you this fall and, above all, what you have to read in the evening at home while relaxing or during your work break.

by **Daniele Brahimi**







OUR FINAL SCORE

» Gameplay 60%
If only it had been smoother...

» Longevity 80%
The first two levels are simple, and then... Makes you want to play it again.

Hokuto no Ken

by Lino, alias Bimbomillo

Hokuto no Ken, known as Fist of the North Star in North America, is a 2D fighting video game produced by SEGA for PSX1 and developed by Arc System Works, based on the series of the same name. The video game was originally released for arcades in 2005 in North America by SEGA.

It was the first Hokuto no Ken video game to feature fully 3D polygonal graphics. The main mode of the game follows the plot of the manga (from chapters 1 to 136) as the player takes control of Kenshiro from the beginning of his journey to his final battle with Raoh. The game faithfully reproduces scenes from the manga, apart from a few deviations and omissions made for gameplay convenience. The voices of the various characters are the same as in the anime, with the exception of Kenshiro and Rei. Although a little blocky, the wonderful and detailed textures make this game a masterpiece for PS1.

The detailed backgrounds and story make for an enjoyable game, which consists of a classic free-moving beat 'em up on screen. The combat phases are interspersed with scenes that faithfully reproduce the story. On some enemies, it is possible to perform a kind of "fatality" by quickly pressing the sequences of directions indicated on the screen. The game was released exclusively in Japanese, but an Italian patch has recently been released that replaces both the text and the opening music. The patch is by Scorpion2k7 and the link can be downloaded from YouTube.

To use the patch, you need to have the game dump in bin/cue imgburn format (no multitrack), which can only be found on the internet in multitrack format.

If you have the multitrack version, Scorpion has included a patch in the archive to convert the game to the correct format (you will find a simple guide inside).

The archive contains two translation patches, the first containing the translation of all texts and intros into Italian; the second containing the translation of texts, intros + an extra menu to select the video output format, some cheats, and the ability to view the game in native 16:9 (both on console and emulator) without altering the image.
I do not recommend using PAL video output because it alters the game's audio.

Translation file link:
https://mega.nz/file/mEdEgThA#cgXI0qmS8G2ykXcrlH6w3UcbuojcGA7IKt4nyfXS_hs

Warning: If you use the patch with the trainer in the initial additional menu, the game may appear to be frozen, but in reality there is a problem with some analog controllers (both on console and emulator). Only in that menu do you need to disable the analog function of the controller. On consoles, press the central controller button; on emulators, you can set a button that enables/disables the analog function. Once you have passed the menu, you can safely enable the analog function.

If you are using Duckstation, you must disable the 'pgxp geometry correction' option in the graphics menu, otherwise the game may crash at certain points (the game is not compatible with this graphics option). We would like to thank Skorpion for this masterpiece of a patch, which makes the game even more enjoyable. If you have any problems with the patch, you can contact him via the link posted on YouTube.

OUR FINAL SCORE

Grafics: 92%
Longevity : 90%
Sound : 93%
Total : 96% awesome.

Speed Freaks

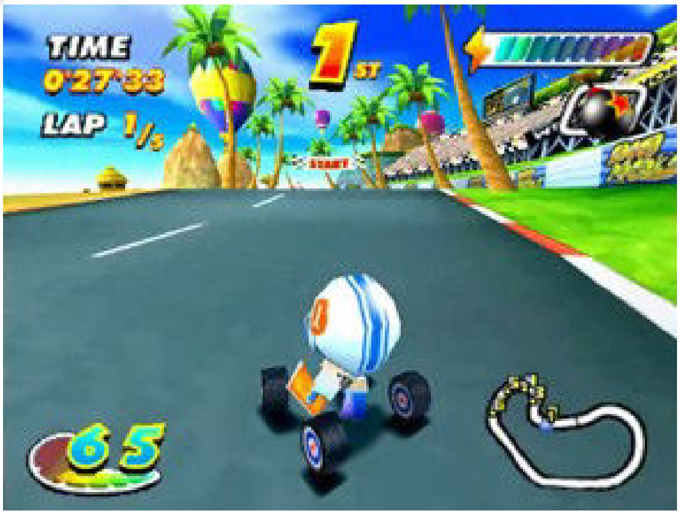
by Lino, aka Bimbomillo

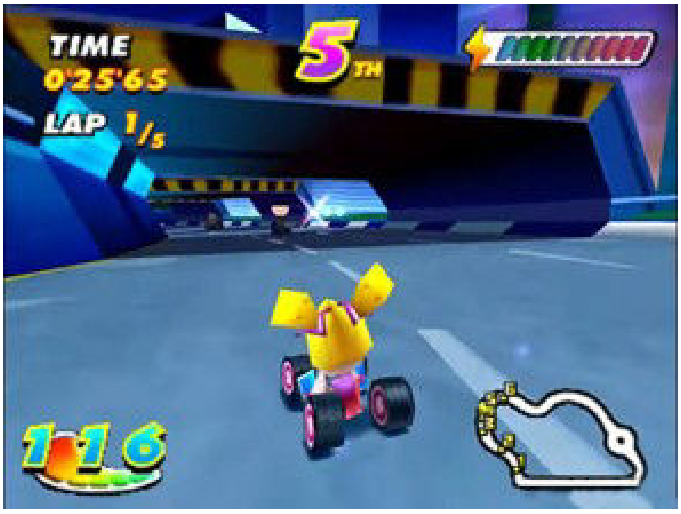
Speed Freaks (released as Speed Punks in North America) is a racing video game developed by Funcom Dublin for PSX1 and released in 1999. It supports up to two players (four with a PlayStation multitap).

Speed Freaks is a kart racing game with several single and multiplayer modes. During the game, players take control of one of nine characters (some are unlocked by finishing the game) and drive karts around tracks with various themes. Races are played from a third-person perspective, with the viewpoint behind the go-kart of the chosen character. The objective of the game is to finish a race ahead of other drivers, who are controlled by the computer and other players, or to complete a circuit in the fastest time. Speed Freaks features nine original characters that the player can choose to compete as or against. Six characters are immediately available at the start of the game, and three more characters can be unlocked by beating them in a challenge. Each character's kart has different capabilities with varying levels of speed, acceleration, and handling.

Icons marked with question marks are deployed on the race circuits, giving weapons or bonuses to a player's kart if the vehicle passes over them. Pick-ups, such as missiles and slime, allow racers to hit others with objects, causing them to spin out and lose control. A kart that obtains the pick-up in stealth mode is temporarily invulnerable to attack. Speed boost token lines are located on the tracks. By running over these tokens, a kart collects them and fills the cells on the speed boost bar. The player can use the accumulated speed boost to increase speed at a faster rate than normal acceleration.

Speed Freaks has three single-player modes: Tournament, Single Race, and Time Attack. In Tournament mode, a player races against five computer-controlled characters in a series of four races called leagues. There are three leagues available, titled Easy,





Medium, and Hard. Initially, only the Easy league is available. Upon completing this league, the Medium league is unlocked, which in turn unlocks the Hard league when completed.

To continue, a league position of third or higher must be achieved in each race. If a player finishes in fourth place, the race must be retried until a third-place finish or higher is achieved. Points are earned by finishing in the top three positions in a race; first to third place receive ten, seven, and five points.

Computer-controlled characters also receive three points for fourth place and one point for fifth place. The racer with the most points after all four races have been completed wins the championship. Winning all races in a championship unlocks one of three challenges. In each challenge, a player races against a computer-controlled character in a single race, where if the player wins the race, their opponent becomes a playable character.

In single race mode, players compete against five computer-controlled characters on a track of the players' choice from those featured in Tournament mode. In time attack mode, players race against the clock through the same tracks featured in Tournament mode, attempting to set the fastest time possible.

Speed Freaks also has two multiplayer modes: Vs. and Multiplayer. In Vs. mode, two players race against each other in a single race or through a league, on the same tracks that are present in Tournament mode, with the screen split in half.

In Multiplayer mode, three or four players race against each other on one of 12 additional tracks. The tracks in this mode have less detail than those in single player and Vs. modes. This game mode requires the Multitap peripheral so that four controllers can be connected to the PlayStation.

Some tracks have shortcuts that allow you to reduce the distance to first place. Some shortcuts are always available (although well hidden), while others open up on one of the three laps available.

The control system is a bit sluggish and it takes practice to master the vehicle. The shoulder buttons allow you to take tighter corners. The music is very nice and suits the game well. The game is a lot of fun and, in my opinion, one of the most entertaining games released on PSX, much better than Crash Team Racing, which is easier to play.

OUR FINAL SCORE

Graphics: 91%
Longevity: 88%
Sound: 92%
Total Score: 92%
Really beautiful and fun.

El Arte Digital: when technology meets creativity

During my after-hours forays into the web, I came across a book published in 2021 by the Colombian publishing house CelerSMS, written by Vladimir Kameñar, an electronic engineer, software developer, and professor at the University of Bogotà, entitled "Demoscene - el Arte Digital". The text is obviously all in Spanish, and so far I have not been able to find a copy in English. So, armed only with my limited knowledge of the Iberian language, but intrigued by the title, I ventured, for now, only into the preface. From it, I gained a timely and objective overview of the subcultural phenomenon known as "demoscene."

For those who have been hanging around home computers since the early 1980s, this is nothing new: the cassette tapes and pirated videogame disks that (just confess!) you used to load onto your beloved ZX Spectrum, Atari 800, MSX, and Commodore 64 were often preceded by so-called "crack demos" or "intros," short programs filled with music, bright colors, and visual effects in which the pirate group on duty boasted about being the first to crack the videogame, often close to the official release date. It was with these small programs, designed to impress videogame users but above all rival groups, that the "demoscene" began, a sort of digital art that evolved rapidly over time, keeping pace with updates in personal computer hardware. From 8-bit to 16-bit systems (Amiga and Atari ST above all) to 64-bit PCs (Windows, Linux, and MacOS), digital works - yes, in many cases they are true works of art - and demoscene productions have reached levels of great complexity, continuing to skilfully and creatively combine text, graphics, animations, musical compositions and ultra-optimised code.

The challenge has always been to reach (and sometimes overcome) the intrinsic technical limits of a system (e.g., the amount of available RAM, the number of colors in the palette), making the most of each processing cycle to reproduce unprecedented graphic and musical effects in real time. The results have often been jaw-dropping. For the beginners: we are not talking about playing a pre-recorded video consisting of a finite number of frames per second (a difficult task for many 8-bit machines), but about running a software program capable of displaying moving images, rapidly rotating graphic elements, text scrolling smoothly in various directions, animations crossing the screen, etc. All accompanied and punctuated by a soundtrack created specifically for the demo.

What's the use, you may ask. None. Does art in general have any declared aim? Isn't it one of the highest forms of human expression? Well, the demoscene, understood as "Digital Art," is no exception.

David "Cercamon" La Monaca

P.S. In the next issues of RMW, we will try to trace the history of this art, which originated mainly in Europe and then spread throughout the world, through anecdotes, stories, and interviews with its protagonists, old and new. Did you know that in Germany and Finland, the "demoscene" has been declared a cultural heritage by UNESCO?

Disclaimer

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

RetroMagazine World is licensed under the terms of: Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) https://creativecommons.org/licenses/by-nc-sa/4.0/

This is a human-readable summary of (and not a substitute for) the license. You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material

The licensor cannot revoke these freedoms as long as you follow the license terms. Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for commercial purposes.

ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

BY NC SA

RetroMagazine World
Year 4 - Issue 29 - WINTER ED. 2025

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

AI Disclaimer *- No content included in this digital magazine (editorials, articles, reviews, any texts or images) has been produced or generated by AI tools*