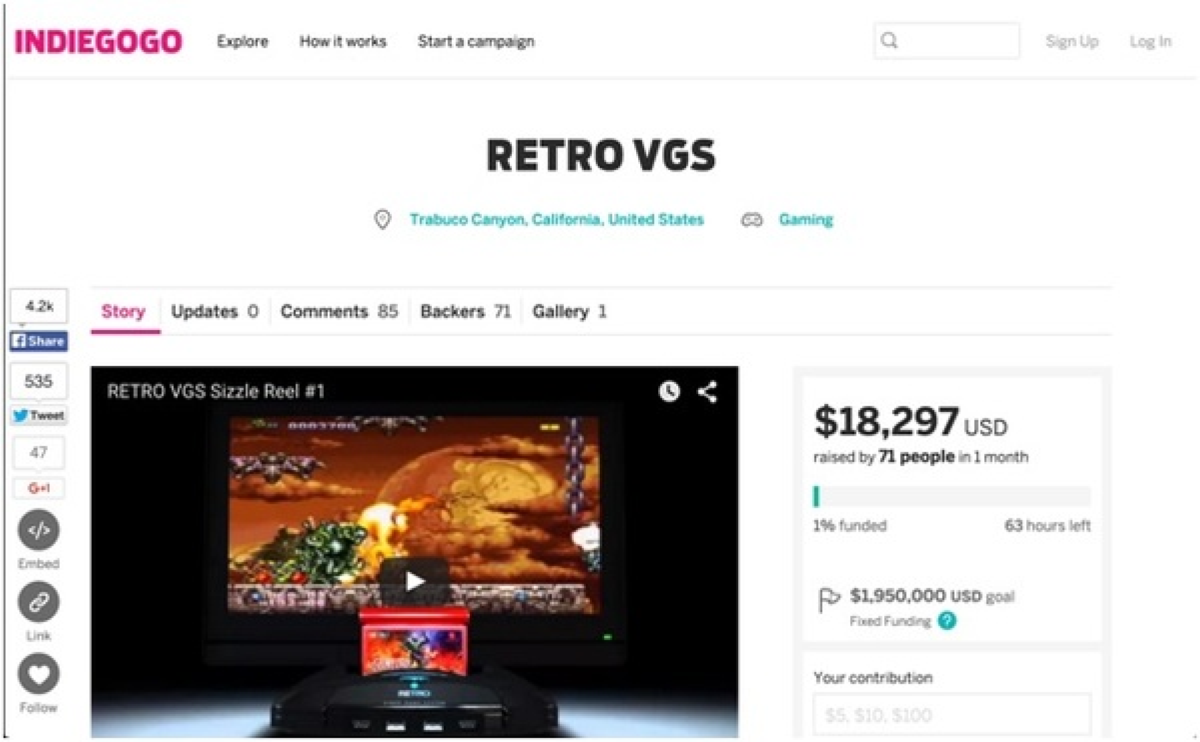
The Modest Fantasy of the PICO-8

pastemagazine.com/articles/2016/01/the-modest-fantasy-of-the-pico-8.html

In early November, the RETRO Video Game System concluded its slow crowdfunding demise on Indiegogo. Conceived by Mike Kennedy, founder of game auction site Game Gavel and the RETRO Video Game Magazine, the RETRO VGS is a new console—or rather, the proposal for a console—that, in Kennedy's own words, "harkens back to the simpler days of gaming where games were fun without all the modern day hassles." Rallying behind that key conceit, the VGS is adamantly and anachronistically cartridge-based, meant not only to shake off the bad habits of the modern game industry but to help preserve its future by emulating its past.

Kennedy aims to revive a bygone era of videogame history, quite literally by exhuming the body of the ill-fated Atari Jaguar console and retooling it with modern hardware. He claims that developers' reliance on network-connected consoles has led to bug-ridden games, a glut of post-release DLC, and an uncertainty about whether our games will continue to function in the coming decades. By fixing our games in ROM, we can return to the halcyon days of error-free, instant-on gaming. But barring a miraculous cash influx or a crowd-funding audible, the VGS will remain as it was born—a 3D-rendered non-object encapsulating the nostalgic impulses that drive so many self-proclaimed "retro gamers."



RETRO VGS

Trabuco Canyon, California, United States · Gaming

$18,297 USD
raised by 71 people in 1 month

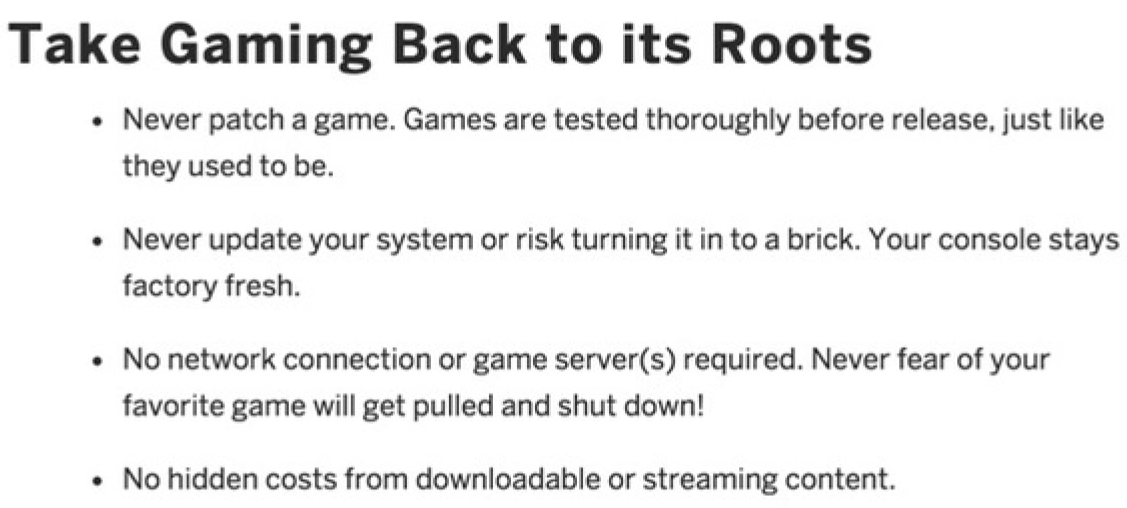
1% funded · 63 hours left

$1,950,000 USD goal
Fixed Funding

Your contribution
$5, $10, $100

The RETRO VGS Indiegogo campaign in its final hours.

Part of Kennedy's misstep is the VGS's attempt to consolidate an entire era of console gaming into a single device. Its FPGA core, for instance, is designed for architectural flexibility, allowing developers to "reprogram" the hardware to spec. In effect, it could model the architecture of a SNES for one game and a Neo Geo for the next. Kennedy likewise promises exterior designs that range from N64-style translucent shells, to "Treasure Series" gold, bronze and silver, to "Legend Series" paintjobs that evoke the garb of classic videogame characters. And the VGS's proposed peripheral support reads like a Wikipedia entry on the history of hardware interfaces: USB, HDMI, RCA, 9-pin DIN, RF, RGB, S-Video, and a bevy of platform-specific expansions. These are ambitious plans that aim to be all consoles for all customers. But by mimicking every platform, the VGS ends up with no identity of its own.



Take Gaming Back to its Roots

- Never patch a game. Games are tested thoroughly before release, just like they used to be.
- Never update your system or risk turning it in to a brick. Your console stays factory fresh.
- No network connection or game server(s) required. Never fear of your favorite game will get pulled and shut down!
- No hidden costs from downloadable or streaming content.

The RETRO VGS's Indiegogo campaign promised to "Take Gaming Back to its Roots."

I understand Kennedy's impulse to recapture an era of games that he finds personally meaningful. Our media histories are deeply personal, subject to the contingencies of culture, class, taste, and commerce. But retro evangelists often conflate the personal with the universal and complain when others' trajectories diverge from—or worse, contest—their own. When Kennedy says the VGS will "reintroduce gamers to the culture of gaming," we should raise a critical eyebrow. Nostalgia is both comfort and curse. We tend to valorize the constructions of our remembered past, so often molded by youthful minds and refracted by imperfect recollection. What do we really mean when we say we want to return to the "simpler days of gaming"? What does that look like? And do we really want what we think we want?

Enter the PICO-8, a self-described "fantasy console" developed by Joseph White (aka zep) of Lexaloffle Games. At first blush, PICO-8 software looks like artifacts of the simpler days that Kennedy romanticizes. Their chunky pixels, limited palette and bleepy-bloopy waveforms are evocative of a range of 1980s consoles and PCs, from the NES to the C64. But the PICO-8 is a product of 2015, a virtual machine purposefully and modestly built, as White explains, "for making, sharing and playing tiny games and other computer programs."

PICO-8's specifications are designed accordingly. Its native resolution is 128×128 pixels—yes, sub-Game Boy—with a fixed palette of sixteen colors. Graphics are efficiently tile-based, split between 128 8×8-pixel sprites and an equal outlay of map tiles (along with an additional 128 tiles shared between both). The two virtual joysticks include four-way directional buttons and two action buttons apiece. There are only four channels for music and sound effects, but there's a nice range of available waveforms (ok, eight), from simple sines to bursts of noise. Finally, carrying the console metaphor to its logical conclusion, PICO-8 games are bundled as 32K "cartridges," meaning your average text document occupies more data than the console's tiny games. While White admits in the PICO-8 documentation that these are "harsh limitations," he assures us that they are "carefully chosen to be fun to work with, encourage small but expressive designs and hopefully to give PICO-8 cartridges their own particular look and feel."

The palette limitations alone generate their own unique aesthetic compared to other modern platforms, but what's remarkable about PICO-8's approach is that it also provides a fully-integrated development environment. All the tools to code, sprite, score, play, share and distribute PICO-8 games are built into the program's diminutive frame. After its brief startup animation, booting PICO-8 greets users with a software anachronism—a command prompt rendered in PICO-8's house style. While modern users might balk at the lack of a GUI, computer users from the '80s (or today's Unix devotees) will recognize this as the central hub from which programs are navigated, listed and loaded. Users may either run commands directly from the prompt or press ESC to pop over to five development views, where they may type code, draw sprites, tile maps, sketch sounds effects or build patterns in the music tracker. The total package feels both hand-crafted and carefully constrained, a homespun nod to the best features of bygone platforms. Even the interface is beholden to PICO-8's self-imposed palette and resolution restrictions, so both development and play feel of a single piece.

The PICO-8 boot sequence and command prompt.

When I asked White about PICO-8's inspirations, he named several key predecessors: "The 16-color palette was partly inspired by the C-64, and the work of many pixel artists who still use it. PICO-8's palette is much more garish though—I spent a lot of time pixelling in EGA as a kid and I think that also had an influence. Also, the audio system has some roots in the 4-channel trackers that emerged on the Amiga 500, but using waveform and noise generators instead of PCM." According to White, other platform inspirations include the Apple IIe, BBC Micro, cellphones, a Famicom played on a blurry CRT, the Atari 2600, and PICO-8's own elder sibling, Voxatron. Much like White himself, who hails from New Zealand but now works in Japan, the PICO-8 has a cosmopolitan makeup that cuts across multiple times and territories.

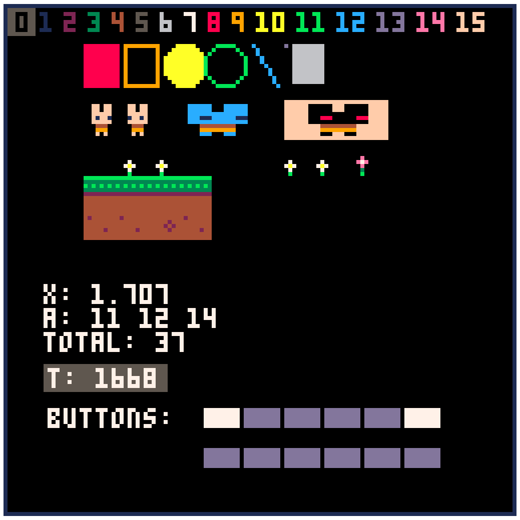
White's software chimera works because he understands where nostalgia ends and the needs of modern development begin. PICO-8's idiom, for example, is a custom subset of Lua, an easy-to-approach language likely familiar to many game programmers, whether they script emulators or use the free 2D engine LÖVE. Though PICO-8 initially ran BASIC to stay true to its '80s pedigree, White switched because "Lua has a small footprint and is portable and quite fast. I think any language could have worked though, as long as it allowed the user to mess about in an unstructured way without worrying about programming style." PICO-8 Lua feels snappy and sensible. There are just enough functions to get up and running quickly without the bulky overhead of complex libraries or object-oriented syntax. Need to draw a sprite? Provide an index and two coordinates. A circle? Coordinates, radius and color. It's fast and friendly for programming beginners.

Drawing a circle in PICO-8.

```
> CIRC(75,75,15,
```

White also designed features to help foster a development community. PICO-8's built-in screenshot and animated GIF generators simplify social media sharing. Likewise, users can post cartridges, whether experimental demos or finished games, to the Lexaloffle forums, where others may either play them for free within the browser or download them to their local PICO-8 folder. This second feature is especially important for young or new programmers. By posting your game online, you share its code and assets with others. Since PICO-8 programs are laid bare by design, tricks and techniques disseminate quickly and encourage the community to poke around, modify, and improve others' work. These features are a modern take on the ecosystems surrounding PC development in the '80s, when users could type in code listings they found in magazines, edit them, and make them their own.

PICO-8's built-in API program demonstrates many of its features.



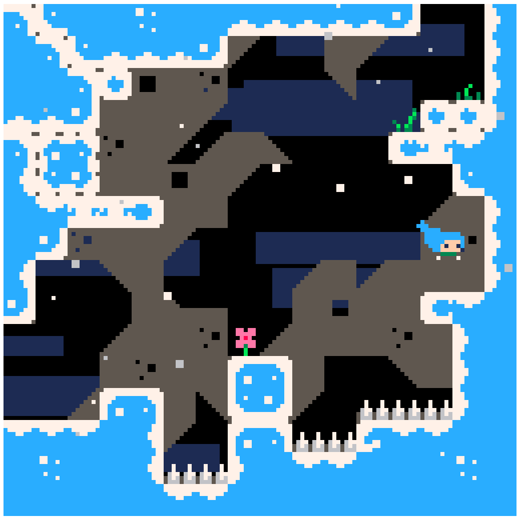
```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

X: 1.707
A: 11 12 14
TOTAL: 37
T: 1668
BUTTONS:
```

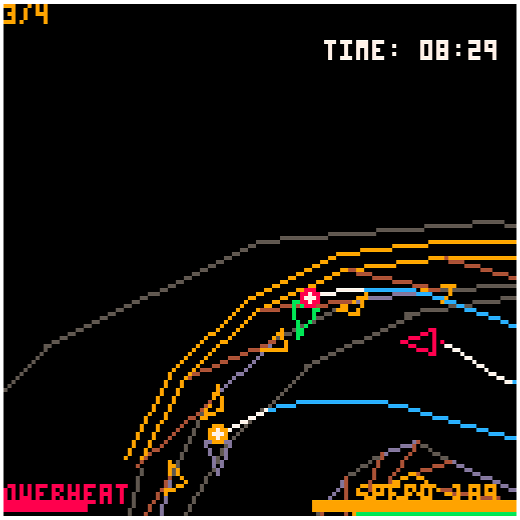
These sensibilities are the root of why White calls PICO-8 a "fantasy console." "Beyond being just a machine," he says, "PICO-8 is also meant to cultivate the surrounding ecosystem that gives a console meaning: development tools, design culture, player community, distribution—things that are all needed to make a console operate as a medium and a platform. So by calling it fantasy console, I want to say: you can get all of that stuff but without the inconvenience of actual hardware." And the community has responded in kind. Beyond the forums, which serve as the nexus for PICO-8 users, there have already been three PICO-8 print/PDF zines, an active #pico8 hashtag, Twitter accounts, web and video tutorials, platform-inspired pixel art, chiptunes, and a PICO-8-specific game jam.

Browse through the forums and you will find a remarkable diversity of games and demos: Matt Thorson and Noel Berry's *Celeste* is a cute-but-tough platformer that evokes both *Super Meat Boy* and *Ice Climber*; Impbox Software's *PICORACER-2048* feels like a *Wipeout* sequel ported to a glitchy Vectrex console; NuSan's *Cyclo 8* manages to cram a seven-level, *Trials*-style, "physics bike" game into a PICO-8 cartridge; and Sophie Houlden's *Dusk Child* is a lovely, evocative puzzle adventure game that taps into the PICO-8's more subdued colors. If chiptunes, tools and Euro-style PC demos are more your speed, you can try out J-Fry's astounding Terrain Renderer, AfBu's weirdly meta CHIP-8 Emulator, Pizza's PIZZAMUSIC music cartridge, oinariman's Shodo calligraphy drawing tool, abrugsch's cheeky 8-bit tape loader, and zep's own graphical demo Bitpools. There are dozens more games and experiments, ranging from arcade-style shooters to inventive demakes to ray casting demos. Less than a year into its life, developers are pushing against PICO-8's constraints, literally PEEKing and POKEing at its guts to wring out results that at first seemed impossible given the platform's limited means.

The PICO-8 cartridge Celeste*, designed by Noel Berry and Matt Thorson.*

It is hard to manufacture this kind of community enthusiasm. Similar tools and platforms arrive and vanish every year with little interest. When I asked White why PICO-8 seems to have struck a nerve with developers, he says, “Perhaps one reason is that it’s easy to see at a glance what the boundaries of PICO-8 are, and to imagine what you might make with it. I’m going to make an 8×8 bunny and a program to walk around a little forest. Modern programming environments and even game making tools tend to be very flexible at the cost of offering an intimidating abundance of possibilities.” A laser focus on platform identity is what differentiates White’s fantasy console from Kennedy’s. White salvaged his favorite bits from the past, set specific parameters that felt sensible and fun, and built an infrastructure around his platform that would engender experimentation, sharing and play. Kennedy tried to salvage an idealized past, set no limits on his dream, and asked a community to foot the bill.





Impbox Software’s glitch racer PICORACER-2048*.*

Lofty rhetoric aside, a costly physical totem is the only marker that distinguishes a VGS game from any other Steam release, while the PICO-8’s virtual cartridges are immediately recognizable, distinct, and most importantly, playable. So why not take the next step and build a PICO-8 console? “Although it was fun to think about what a real PICO-8 might look like,” White writes in the first PICO-8 zine, “I never felt it would benefit from having an official physical form. Choosing specs was more about encouraging a certain design culture and development experience rather than being realistic or plausible.” As cultural critic Svetlana Boym reminds us in *The Future of Nostalgia*, there is an alluring ambivalence that haunts our longing for the past. Nostalgia, she writes, “is about the repetition of the unrepeatable, the materialization of the immaterial.” When we reflect on what we really want from the past, it’s an evocation, not an invocation. The PICO-8, while virtual in its implementation, is a fantasy made real.

Nathan Altice is a writer and artist who lives in Richmond, VA. He published his first book, I AM ERROR*, a platform study of the Nintendo Entertainment System, in May 2015. He writes at* metopal.com *and complains* @circuitlions*.*