

Chip8/vision8

Pioneer

Vision8 emulates a pioneering cross-platform virtual machine. Chip8 was introduced in 1975, on RCA CDP1802-based computers, like the RCA's **TELEMAC 1800** home console, bundled with games on cassette tape, and the **COSMAC VIP**, advertised to UK hobbyists as a home computer into the 80s.

Chip8 games were played with a hex keypad or joystick, using chunky mono graphics reminiscent of early TV games. Besides various puzzle games, *Pong* and *Brix* revisit *TeleTennis* and *Breakout*, each in under 300 bytes, with on-screen scoring. *UFO* is a slimline *Space Invaders* with a base.

Vision8 runs on Unix systems with X in 8, 16 or 32-bit per pixel modes, MSDOS with CGA, 8 bit MSX, ColecoVision and Adam micros, and 32-bit AmigaOS 3 systems. Its portability is exemplified by recent ports to GameBoy Advance and HP48 calculators! I renamed the variable **sync** in *X.c* to get *Vision8* to compile with **GCC**. Renaming it **synch** in three places resolved a clash with the GNU function that ensures disk contents are flushed to drive hardware.

Vision8 is small and elegant, though the appeal is minimalist. With its handful of opcodes and tiny programs, the gameplay to program size ratio beats many modern CD and DVD releases.

« **F1** calls up the debugger and numeric keys toggle sprites, 4 background graphic layers, and six sound channels. Audio volume is also key-controllable, though there's no GUI or menu system.

The nicest feature is the way you can stretch the screen to almost any size by dragging the corner of the window as the emulator runs – after a moment the display will resize to fill the new window, smoothly scaled to fit, with no obvious stepping of pixels. It even allows views to be scaled down from the normal GBA resolution, but only steroidal WAP telephonists are likely to benefit from that. Substantially stretched windows grew slow on my K6/500 unless I used **-uperiod** to skip screen updates.

VisualBoyAdvance also runs programs for earlier GameBoys, like the real thing unlike other Advance emulators. It uses SDL on Linux and Windows, requiring a 500 MHz PIII for full speed. My AMD K6 delivered playable results, but sound was uneven and it struggled to reach 90% performance on Advance emulation. Plain GameBoy and GBC support is far less demanding.

VisualBoyAdvance

I tested version 0.8, supplied as an x86 binary with a small, well-commented configuration file, and brief readme text. Numeric pad keys are said to indicate directions but I found that cursor arrows worked, in practice.

Keys are definable by changing the SDL keycodes in the '.cfg' file. This can also select digital joysticks or pads, one to four times zoom or full screen display, half a dozen video filters, and frame skipping in Advance and standard GameBoy mode, from 0 to 5



Lego Racers 2 on VisualBoyAdvance.

frames. MMX instructions and a real GameBoy BIOS may be switched in, but are not required. Border and status displays are also configurable, along with the paths used for ROM loads, screen and state saves, and storage of 'battery-backed' data.

Function keys 1 to 10 save game states, with shift to reload, so you can save and restore ten positions without an external script to shuffle multiple saves. **F12** grabs a screen. **Alt-1** to **4** toggles auto-repeat on the A, B, left and right buttons, mapped to **Z**, **X**, **A** and **Z** (again?!). – at least according to the README, the fourth one should say **S**.

Enter mimics Start and **Backspace** is Select. **Ctrl-R** resets and **Ctrl-P** pauses the emulator. **Esc** quits and **F11** enters the debugger, undocumented but with adequate online help. This offers ARM and Thumb code disassembly, breakpoints on access or write, single-stepping and facilities to view and edit memory, ARM and I/O registers.

Comparisons

Boycott Advance runs some ROMs that VGBA rejects or renders blankly, like the mode 7 Doom scrolling demo, but VGBA's **-nocrc** switch reprieves some with incorrect ROM checksums – another area where VGBA sets out to be strict. *Boycott's* **-romwrite** switch gets some demos running that would fail on a real Advance. *Boycott* suffers from graphics glitches, especially on text, and *VisualBoy Advance* could do with better sound and documentation, but all these emulators work well on reasonably modern PC hardware.

PC Engine emulators

Long-running 16-bit consoles

PC Engine 16-bit game consoles, also branded TurboGraphX, were made in Japan by NEC until the early nineties, strongly supported by Hudson, now Nintendo partners. Mostly rôle-playing and shooting games were produced for it, from 1985-99. *VPCE* and *Hugo* are the two PC Engine emulators for Linux. Both work well, after tweaks, but *VPCE* gives most insight into the Engine.

Hugo uses *Allegro* for platform-independent sound and graphics. The docs are aimed at Windows users, but have a brief Linux section. A bundled *Pong* game helps you prove that it

works, but installation is tricky even though the x86 code comes ready-built. The archive has incorrect permissions so the executable needs 'chmod' to bring it to life. You also need to copy a data file:

```
mv hu-go!l.dat /etc/hugo.dat
```

VPCE, the *Virtual PC Engine*, comes pre-compiled as Intel code for *libc5* and *glibc2*. Each archive contains a small readme file and executables that run in a X window or full screen via *SVGAlib*.

To run the *glibc2* version on my Debian

```
ln -s /usr/lib/termcap.so
```

```
/usr/lib/termcap.so.2
```

That made a link from the installed

terminal capabilities to the specific version requested by *VPCE*. Other versions bombed with a Segfault. Games are not bundled but *VPCE* unzips ROMs automatically, & runs *Hugo's Pong* demo.

The normal screen mode uses 256 by 240 pixels at 60 Hertz, but **-mag n** zooms the screen **n** times in X and Y and **-frameskip** skips frames for slow hosts. *VPCE* defaults to 8-bit colour but shifts to 16-bit if that's your XFree86 setting; otherwise **-truecolor** sets other depths.

Sound uses '/dev/audio' for 8-bit µlaw, unless you specify **-dsp**. As with *Hugo*, I found it rather grainy. Move around with

cursors, a joystick or joystick, via '/dev/js0'. **C**, **X**, **Return** and Space keys emulate console buttons, and **Alt-F3** saves a TGA screenshot. Press **Esc** to quit, twice if in **-debug** mode.

The **-debug** command option invokes a fine debugger. A VRAM browser identifies background, map, palette and sprite graphics. The monitor terminal can disassemble code for the 68516 processor, as used in Apple GS and SNES systems. Digit keys **1** to **4** generate interrupts. You can step, skip, insert NOPS, RETs and RETIs, review memory or enter 'R' to run.

PlayStation

Sony's PlayStation (PSX) game system needs little introduction. It's a CD-based console with a 33MHz MIPS R3000 processor, SPU audio engine, and capable though obsolescent 3D graphics. It has sold tens of millions and still thrives, eight years on.

There are six PSX emulators for Linux; most share a plugin system, so the emulator contributes the GUI, CPU and memory emulation core, and you mix and match add-on code for sound, graphics, CD, pad and network support to suit your host. Emulators and plugins are typically available for Windows and x86 Linux systems, so the name alone does not guarantee a GNU-friendly binary. All the emulators rely on a copy of Sony's ROM BIOS from a real PSX; the US version *scph1001.bin* gives best compatibility.

DIY PlayStation software development is now feasible using open source software plus an *Action Replay* or similar cheat cartridge, patched BIOS, and an ISA card with a bi-directional parallel port. Our links point to penguin-friendly PSX development sites; emulators are a big part of this scene.

PSX Emulators

PCSX development is Windows-led, but the Linux version is at 1.2 and capable, if a little flaky. Console messages warn of GUI bugs while setting up and it rejected some the plugins I tried.

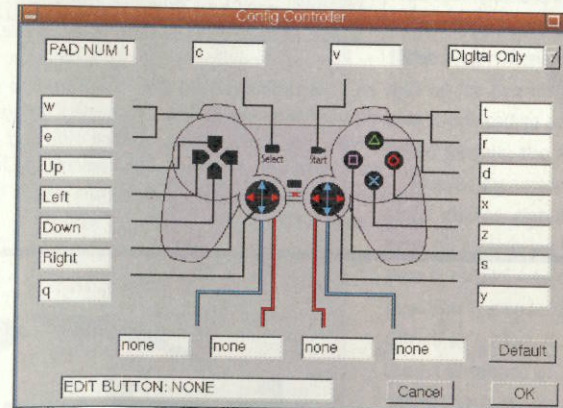
ePSXe also runs on Windows, but the Linux version is slick, solid and feature-rich. Sadly the binary is Intel-specific, whereas *FPSE* runs on Windows and PowerPC AmigaOS, so it is not hard-wired for little-endian systems, though BeOS and Linux versions have vanished from the home page. *GPSE* is a fork of *FPSE* 0.08 made under GPL, with the agreement of *FPSE's* main author.

SOPE development stalled late in 2000. Version 0.04 was released under GPL and has dynamic compilers for Alpha and x86 and interpreters for other systems. You may encounter two other contenders, though work on them seems to have stopped. *PEX* was developed in 2000 with Slackware, but the site was down each time I tried – the emulator seems to have vanished.

PSX tests

My tests used Pete's *SoftX driver1.52* and the *Peops* audio driver 1.1 for OSS. Audio has a configuration GUI, but graphics options are set in a '.cfg' file.

I found cover-mount game demos a cheap and effective test



This *ePSEe* dialogue maps PlayStation buttons to Linux keys.

of compatibility. *ePSXe* was much the most capable running games like *Driver*, *Gran Turismo 2*, *TOCA*, *Tomb Raider TLR*, *International Track and Field* and *FIFA 2000* playably, though at frame rates between 10 and 20 rather than the 50 or 60 they were made for. 3D

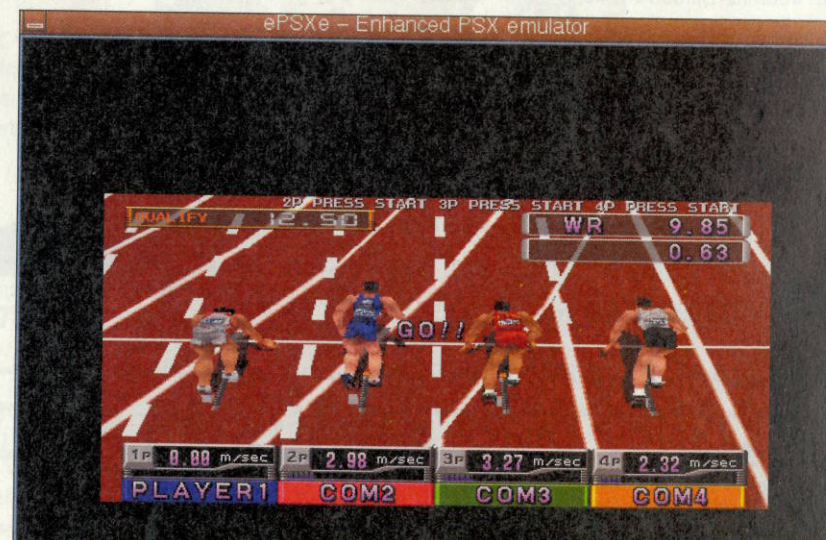
hardware and a modern CPU will close this gap – if you already play 21st century action games, your system should cope fine. The only unplayable game was *Metal Gear Solid*, which flickered to the point of uselessness on my setup.

Streaming CD audio sounded choppy, but in-game effects sounded fine. The graphics plugin saves screens in bloated BMP format, at 1.5MB a time, whereas *XV* packs the same images well within 100KB as PNGs.

PS2 prognosis

Sony have since moved on to the 128-bit PlayStation2 (PS2). You're not going to emulate the PS2 on any desktop computer for a few years yet, but *Pcsx2* might give useful insights into the architecture. A more practical option is Linux for the PS2. LXF

Konami International Track and Field literally runs on ePSXe.



Links

Advance demos: www.gbadev.org

Boycott Advance: <http://boycottadvance.emuunlim.com/>

Emu series online: www.simon.mooli.org.uk/LXF

EPSE home: www.epse.com

FPSE home: <http://fpse.emuunlim.com/>

GPSE CVS: <http://savannah.gnu.org/projects/gpse/>

Hugo home: <http://fpse.com>

VisualBoy Advance: <http://vboy.emuhq.com/>

VPCE home: <http://emuunlim.com/hugo/>

Intellivision extras: www.intellivisionlives.com

PCSX home: www.pcsx.net

PSX emu UK: <http://linux.psxfanatics.com/>

PSX Tux devkit: www.psxdev.de

Simple Direct Media Layer: www.libsdl.org

SOPE home: <http://sope.sourceforge.net/>

VGBA home: www.komkon.org/fms/VGBA

Vision8 home: www.komkon.org/~dekogel/vision8.html

VisualBoy Advance: <http://vboy.emuhq.com/>

VPCE home: www.geocities.com/SiliconValley/Way/3340