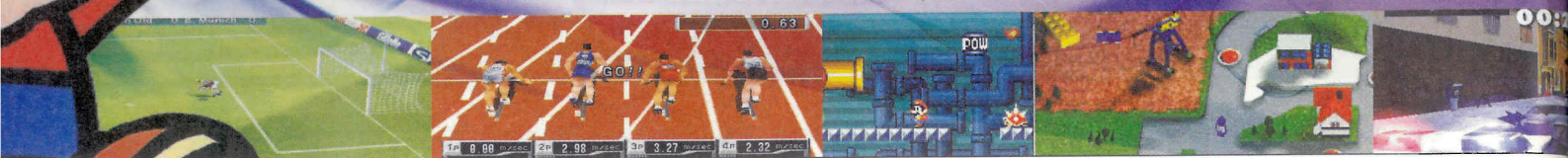




CONSOLE cavalcade



Simon Goodwin rounds up of stray console emulators covers Playstation, Intellivision, Lynx and GameBoy Advance.



On the CD and DVD
Your coverdisc holds a dozen freely-distributable Advance demos, including *FruitLand*, with source for the GCC cross-compiler, though its makefile presumes MSDOS paths, and Linux emulators for PlayStation, PC Engine, GameBoy Advance, Lynx, Chip8 and Intellivision.

This is our last look at game console emulators in this series, taking in attempts to bring PlayStation, PC Engine and Intellivision software to Linux, plus emulators for Atari's Lynx and Nintendo's current GameBoy Advance. And if the Intellivision is not retro enough, try Chip8, an early cross-platform virtual machine which makes even that look high-resolution. Next month we return to home computers.

Handy Atari

Handy is a free emulator for the Lynx colour hand-held computer, developed by former Amiga staff, funded by Epyx and eventually marketed by Atari. Good games, stereo sound and a colour LCD were Lynx strengths, but meagre marketing and battery life limited sales, though they're no obstacle to emulation.

Handy comes with HTML documentation and automatically unpacks 'zip' and 'gzip' ROM files. It uses *SDL*, the *Simple Direct media Layer*, for hardware abstraction. It supports sound, scalable 16-bit colour graphics, joysticks, screenshots and game saves.

Your X display must be in a 16-bit colour mode. *SDL 1.2.2* or later is recommended, and the *libz* compression library which is standard on most Linux distros. The emulator comes precompiled for recent Linux systems – it called for *libstdc++-libc6.2-2.so.3*, which is not part of the Debian 2.2 distro, but a link to *libc6.1-1-2-1.9.0.so* got *Handy 0.82R1* running.

Graphics are fast, accurate and system-friendly, though sound was rather jerky with the screen scaled up to the maximum X window size, using 16 pixels for each original. Sound and vision were smooth and larger, though still only filling about a fifth of my monitor, in *SDL's -fullscreen* mode.

Besides game cart images you need a copy of the small Lynx ROM bootstrap, which is not bundled for copyright reasons. Many games expect four joystick or pad buttons in addition to directional control, though you can use **Z**, **X**, **1**, **2** instead, with cursor arrows on a keyboard, plus **F10** to reset and **Enter** to pause. The Lynx facility to rotate the screen left or right, vital for games like *Klax* and *Gauntlet*, is triggered with **F11** and **F12**, while the **Home** key resets the normal view. Initial **-scale**

parameters from 1 to 4 adjust the emulated display window size. **F3** and **F4** save and restore game state, **F9** saves up to ten snapshots of the screen, and **Esc** quits the emulator.

Intelligent TV?

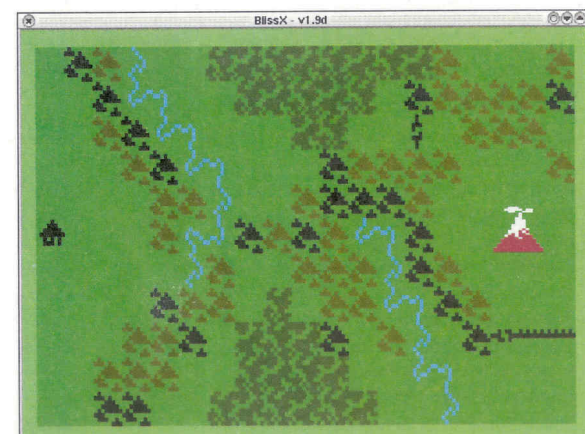
Mattel's Intellivision console provided stiff competition for Atari's VCS two decades ago. The CP1610 central chip is a true 16-bit processor though its CISC instructions are assembled from ten bit sections. Sound comes from a three channel GI beeper later popularised in Spectrum 128 and Amstrad CPC home computers.

There are two Intellivision emulators for Linux. *Bliss* is best for the casual x86 gamer and former user, while *jzIntv* gives greater insight into the depths of Mattel's machine and is more portable.

You need ROMs from the original system, in addition to game cartridge ROMs: the small graphics *grom.bin* and the kernel *exec.bin*, which loads at \$1000, with games at \$5000. Some games also require *ecs.rom*, and Intellivoice speech synthesis demands *ivoice.rom*.

Bliss

Bliss comes in versions for Java and X, with a perfunctory Readme file. The X version, built on *SDL 1.2.3* and *Zlib 1.1.3*, is sluggish so the Java one will try the patience of all but the catatonic or bleeding edge Virtual Machinists. At first *BlissX* sniffed at my Debian 2.2 system because it required



Intellivision *Dungeons and Dragons* zoomed up on X by *Bliss*.

libSDL_image-1.2.so.0, absent even after building *SDL 1.2.3* from source. *libSDL_image* is an uncommon requirement, packaged apart from the main *SDL* binary and source collections; it's not in the stable Debian distro, but will be in 3.0. *Bliss* key bindings are fixed, with digits for the calculator grid of the original Mattel keypad, **Del** for clear and **Z**, **Ctrl** and **X** for the three action buttons. **Enter** and the arrow keys complete the control set.

jzIntv, Joe Zbiciak's *Intellivision Emulator* – better documented than *Bliss*, and less fussy about *SDL*, though linking may require

GameBoy Advance

Nintendo's GameBoy Advance is a modern hand-held console running 240 by 160 pixel 2D colour graphics, 384K of RAM in three sections, and games on ROM cartridges, typically a few megabytes each. The British-designed ARM7 RISC processor is clocked at 16MHz, fast enough to offer interesting gameplay without outflanking interpreting emulators on a modern desktop Linux system.

The Advance offers four sampled sound DMA channels as well as GameBoy beeps. The display supports sprites with 16 or 256 colours, hardware scrolling, tile flipping and rotation, and a high colour display, souped up from the Super Nintendo console.

An eight-bit Z80 coprocessor lets it run older GameBoy software. My *LXF21* column listed ten emulators for such games and demos, so GBA emulation would be the long way round, but the new handheld has been adopted fast by emulator writers as well as gamers; there are already three competent Advance emulators for Linux.

It's feasible to develop Advance software on your own, making this a good route into the games industry. High ROM prices and low margins leave a niche for freelance programmers familiar with classic games and *GCC*. Demo sources, tools and examples abound, though you'll need backing from a Nintendo licensee to get your game into the shops.

Boycott Advance

Boycott Advance is beerware from Niels Wagenaar of The Netherlands, who'd appreciate a postcard if you can't send beer. The release version is not crippled, so registration just gets you (and Neils) a warm feeling, access to pre-releases and your name in the credits.

I tested version 0.22 of the *SDL* remix. Base requirements are a 500MHz x86 with 64MB RAM and 16-bit colour. It managed about 45 frames per second on the demo game, a neat revamp of *Pong*, in the default tiny X window. This rate falls to 15 to 20 in the top scale setting, but remains playable with a frame skip of 3.

Emulation of the 32-bit ARM processor, rather than the sound or graphics, makes Game Boy Advance emulators greedy for CPU time, but the X overhead is substantial; after selecting **-fullscreen**, *Boycott* delivered smooth graphics at 70 Hertz, outrunning the nominal 60 Hertz LCD, on a custom console the same size as the **-scale 4** window. The switches must follow the command and ROM name, peculiarly.

Four adjacent keys, **Z**, **X**, **C** and **V** mimic left, A, B and right buttons respectively, with **Enter** to start, **Tab** for select, **F10** to reset and **Esc** to quit. Arrows, joypad or joystick indicate directions, while **F9** grabs a screenshot, **F1** pauses and **F2** continues. **F3** and **F4** control the frame rate display in the

makefile tweaks to suit your library locations. It's released under GPL and also suits MacOS, so it's not tied to Intel architecture.

The documentation includes a well-written CP-1600 programming guide, aimed at people who know nothing more than a little BASIC. The overview of the source and utilities is particularly helpful, though rather old. You get utilities to convert ROM images between 10 and 16-bit format and a CP-1600 code disassembler. The ESD-based sound is CPU-intensive but accurate, mixing 16-bit data at sample rates from 4 to 48KHz.

window border, and **F5** and **F6** trim the update speed, though they respond so fast *Boycott* often skips from one in eight down to 1:1 or back again each time you press the key, requiring several tries to chance upon any intermediate setting.

VGBA

VGBA is the latest from Marat Fayzulin, author of fine MSX, NES and earlier GameBoy emulators. I tested the pre-compiled version 1.2 which unpacked and ran readily on my Debian 2.2 system.

VGBA comes with HTML documentation cloned from Marat's other emulator docs. The recommended minimum host is a 400 MHz PII with fast graphics. **-sync** can limit fast machines to accurate speed by tying the emulator to your screen refresh rate.

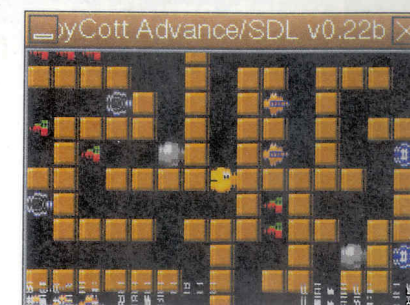
You must run your X display in a 16-bit mode as the console uses 32K colours. Version 1.2 is a major update with much better sound and interrupt handling than previous *VGBA*s. Prototypes ran on MSDOS, but lacked many features. The Win32 version is commercial so *VGBA* is not open source; ports are available for several Unix architectures, but the code relies on a little-endian host CPU, ruling out versions for SPARC, 68K or PPC. The sound driver uses Posix threads, standard on Linux.

VGBA runs most commercial GameBoy Advance ROMs, but mainly addresses game developers and demo programmers who can't afford Nintendo's devkit but want to impress potential publishers. To avoid copyright problems the Nintendo BIOS is emulated by trapping SWI instructions. This is less hardware-compatible but faster than running the original BIOS, emulating right down to the silicon. It's only likely to fail on demos that break Nintendo development guidelines by poking the hardware directly.

The main control keys are **Space**, left **Ctrl**, **Tab**, and **Enter**, but sundry letters are polled simultaneously so it's easy to find an ergonomic layout. Function keys save and load state in '.STA' files, toggle autofire on the four main buttons, reset with **F11** and quit with **Esc** or **F12**. You must quit to load a different ROM.



Blues Brothers for the GameBoy Advance emulated courtesy of Titus and *VGBA*.



GameBoy example *FruitLand*, shakily rendered by *BoyCott Advance*.

