(https://www.libretro.com/)

HOME (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/HOME-2/) · LAKKA (HTTP://WWW.LAKKA.TV) · LIBRETRO (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/API/) · RETROARCH (HTTP://WWW.RETROARCH.COM) · BLOG (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/CATEGORY/BLOG/) · MISSION (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/MISSION/) · DOWNLOADS (HTTPS://RETROARCH.COM/?PAGE=PLATFORMS) · NIGHTLIES (HTTP://BUILDBOT.LIBRETRO.COM/NIGHTLY) · DOCUMENTATION (HTTPS://DOCS.LIBRETRO.COM/) · FORUM (HTTP://LIBRETRO.COM/FORUMS/) · DISCORD (HTTPS://DISCORD.GG/C4AMCEV) · DEVELOP (HTTPS://DOCS.LIBRETRO.COM/DEVELOPMENT/LIBRETRO-OVERVIEW/) · CONTRIBUTE (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/CONTRIBUTIONS/) · TWITTER (HTTPS://TWITTER.COM/LIBRETRO) · GETTING STARTED (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/GETTING-STARTED-WITH-RETROARCH/) · POWERED BY LIBRETRO (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/POWERED-BY-LIBRETRO/) · WEB PLAYER (HTTPS://WEB.LIBRETRO.COM/) · PATREON (HTTPS://WWW.PATREON.COM/LIBRETRO) · LOBBIES (HTTP://NEWLOBBY.LIBRETRO.COM/) · PRIVACY POLICY (HTTPS://WWW.LIBRETRO.COM/INDEX.PHP/PRIVACY-POLICY/) ·

RetroArch 1.14.0 release (https://www.libretro.com/index.php/retroarch-1-14-0-release/)

December 13, 2022 (https://www.libretro.com/index.php/retroarch-1-14-0-release/)

RetroArch 1.14.0 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

NOTE: Check our Libretro Core Progress Report (May to December 2022) article here (https://www.libretro.com/index.php/state-of-libretro-core-updatesprogress-report-maydecember-2022/).

Changelog

1.14.0

- AUDIO/COREAUDIO/APPLE: Allow coreaudio3 driver to work with audio devices that have 2 or more output channels
- CHEEVOS: Fix construction of Cheevos badge path
- CLI: Fixed not getting any output when running –version or –features without –verbose
- CLI: Fixed crash when running empty – parameter (it proceeded to content loading)
- CLI: Reformatted –features to require less rows and to be more consistent
- CLI: Added -V shorthand for –version
- CLI: Tab removal + whitespace nits
- CONFIG/MIDI: Prevent MIDI startup error with old configurations
- D3D11: Fix when using shaders with TATE mode arcades etc
- D3D12: Fix when using shaders with TATE mode arcades etc
- D3D12: Added support for break on errors (development aid – define DEVICE_DEBUG to use)
- D3D12: Added support for DRED (device remove extended data) (development aid – define DEVICE_DEBUG to use)
- D3D12: Made D3D12 viewport and scissors to behave more like Vulkan drivers (or be more correct)
- D3D12: Fixed some shaders not appearing with the d3d12 driver – this reintroduces a validation error though but it seems to work as in not crash and is the old RA behaviour. Guess we need a different way of fixing the validation issue.
- D3D12: Fixed validation error on start up due to buffers not being setup correctly for one frame
- DATABASE/EXPLORE/VIEW: Bugfix – RGUI did not clear thumbnail on non-playlist items such as Save and Delete menu_explore_get_entry_playlist_index() returns -1 on invalid entries, but the variable where it was stored was unsigned
- DATABASE/EXPLORE/VIEW: Bugfix – XMB+Ozone cleared thumbnail in Quick Menu when navigating away from Run
- DRM/ODROID GO2: Implement get_video_size for DRM GL context driver

- FASTFORWARD: Restore framelimit on fastforward toggle. Fast-forward was broken after toggling vrr_runloop off, since it will force frame limit to 1.0 (even on every frame) and never restores it. So let's make sure the wanted ratio is applied when toggling FF (Fastforward).
- FFMPEG CORE: Fix runtime error in FFmpeg core when build with FFmpeg n5.1.2 and OpenGL ES
- GFX/VIDEO FILTERS: (picoscale_256x_320x240) Added snn function to upscale Fuse (ZX Spectrum) core borderless output to 320×240. ZX Spectrum resolution of 256×192 was previously unsupported.
- HOTKEYS: Further reorder internal hotkey items for consistency and removed SEND_DEBUG_INFO, OVERLAY_NEXT and OSK from visible hotkey bind list. "Send Debug Info" stuff is removed as much as possible without breakage due to translation files.
- INPUT/AUTOCONFIG: Disable 'pause on controller disconnect' by default – was enabled by default on 1.13.0
- INPUT/MENU: Device Index menu refactor
- INPUT/OVERLAY: Fix analog drift blocking touch input (could occur on overlay_next if physical inputs shown on overlay)
- INPUT/OVERLAY: Fix overlay_next buttons lighting up in unison
- INPUT/OVERLAY: Skip meta keys in input_overlay_add_inputs (not supported by input_state_internal)
- INPUT/WINDOWS/WINRAW: Fix mouse position when using input overlay with mouse cursor
- INPUT/WINDOWS/WINRAW: Fixed mouse position to use the same method required for menu items and pointer when simulating input overlays with mouse, since it won't work with multi mouse method
- INPUT/WINDOWS/WINRAW: Fixed passing mouse position to core also when using aforementioned method
- LEAPFROG: Add Leapfrog (LFx000) Target
- LOCALIZATION: Updates
- LOCALIZATION/INPUT/IME/MENU/ONSCREEN KEYBOARD: Extended IME and Korean OSK
- MENU: Cleanup of help texts
- MENU: Allow toggling info off with the same button
- MENU: Allow menu wallpaper/background reset. Let's also remove the current wallpaper from the screen when pressing Start.
- MENU: Null driver shows with different color (Added for all menus the ability to show "disabled" items with a muted color)
- MENU/DRIVERS: Menu driver first, Audio Resampler removed because it is enough to exist under audio settings
- MENU/INPUT: Moved "Confirm Quit" to Input menu
- MENU/INPUT/HOTKEYS: Input hotkey menu completely overhauled to keep related entries together, and also adjusted some labels and sublabels
- MENU/OVERLAY: Fix overlays behind menu without core running. "Show Overlay Behind Menu" is currently broken with Ozone and XMB (with any other color theme than Plain) when running without a core.
- MENU/MATERIALUI: Fix home screen on first startup – no more stray entries
- MENU/OZONE: Allowed drawing sidebar and thumbnail bar background color also when core is running
- MENU/OZONE: Stopped using different padding and position for savestate thumbnails vs imageviewer
- MENU/OZONE: Removed gradient background effect when core is running, because some themes already have gradient background, which creates ugly rough steps
- MENU/OZONE: Fixed "Gray Light" theme from using the same background as "Gray Dark", which makes selection cursor near impossible to see
- MENU/OZONE: Some whitespace corrections
- MENU/UX: Extend OFF menu value colors
- MENU/UX: Menu icon improvements – Menu Visibility icons (Quick Menu + Settings)
- MENU/UX: Menu icon improvements – Playlist Manager icons
- MENU/UX: Menu icon improvements – Explore icon as database icon
- MENU/UX: Menu icon improvements – View and filter icons as cursor icon (folder icon in GLUI)
- MENU/UX: Menu icon improvements – View save + delete icons
- MENU/UX: Menu icon improvements – Moved Explore + Views below Standalone Cores
- MENU/UX/OZONE: Removed icons from menus where others items don't have icons, and added icons to menus where the rest have icons
- MENU/UX/OZONE: Changed the way "no icon" is handled from kludgy way of not drawing SUBSETTING icon
- MENU/UX/XMB: Changed playlist entry index positioning to bottom right when thumbnails are in vertical mode, because big lists will overlap with arrow and current "breadcrumb" icons when the position is next to current selection
- MENU/UX/XMB: Added a rather nasty hack to prevent showing wrong icons under Explore as "breadcrumb" icon
- MENU/UX/XMB: Optimized certain icon drawing loops (Main horizontal icons were looped even when not visible, and all previous "breadcrumb" icons were looped when only one certain was needed)
- MENU/UX/MATERIALUI: Fixed showing icons where there should not be any (Waitable Swapchains, Show Recording + Streaming)
- OSX/MACOS: Fixed Cocoa keyboard not allowing to map Analog stick
- PS2: Use the recently created ps2_drivers which makes easier the loading and init of all the drivers: Memory Card, USB, HDD, Audio, Controllers
- PS2: Adds exFat support for USB, and probably solves some unexpected issues when using an HDD driver for booting cores/games.
- SDL GFX: Fix no menu on start/blank screen issue.
- SRAM: Don't init SRAM saving without content (gets rid of the redundant logging)

Blog (https://www.libretro.com/index.php/category/blog/)

State of Libretro Core Updates/Progress Report – May/December 2022 (https://www.libretro.com/index.php/state-of-libretro-core-updatesprogress-report-maydecember-2022/)

December 13, 2022 (https://www.libretro.com/index.php/state-of-libretro-core-updatesprogress-report-maydecember-2022/)

We will cover all core updates here from May 2022 to December 2022. It has been a long time since our last core update report, so bear with us as we go through this long laundry list.

There might be more cores that have seen updates which have gone underreported during this time period. We will make an effort to update it as we go along in case we notice any omissions.

MojoZork

- New core, available for Linux, Windows, macOS and Android first

Nestopia

- Allow loading external XML database with baked in database as fallback
- Update and streamline Core Options

NXEngine

- Fixes music no longer working on Windows with recent versions of GCC – and fixes crashes
- Correct aspect ratio

mGBA

- Update core to 0.10.0

FBNeo

- Update to the latest version

melonDS

- Fix frontend language not changing DS language properly
- Add input bitmask support
- Core options changes + update to v2
- Change screen gap option back to 0-192
- Screen gap reduce from max 192 to 126
- Adds Libretro Cheat Code Support so users could add emulator cheat code on top of RetroArch cheat code

Virtual Jaguar

- Jaguar games can change resolutions during gameplay.
  Previously, the emulator only ran at the default 320 x 240 resolution.
  This PR fixes this, allowing other resolutions to display properly.

  This should fix issues in (at least) the following games, based on my testing :
  * Atari Karts
  * Club Drive
  * Evolution Dino Dudes
  * Flashback
  * I-War
  * Pinball Fantasies
  * Trevor McFur in the Crescent Galaxy

Caprice32

- RMP: Knight Lore, 1942, 1943, 3D Grand Prix, Abu Simbel Profanation, Barbarian, Boloncio, Boulder Dash, Bruce Lee, Ghostbusters, Harrier Attack, Match Day II, Myth, Oh Mummy (RMP WIP)
- CORE: Keyboard Transparency Option added
- DC/DB: Compute hash from M3U too
- DC: Implemented replace CMD, now you can load DSK/CDT from UI
- CORE: Fix RPI 1-B (old model)
- RMP: Target Renegade
- CORE: joystick fixes
- CORE: evercade support
- DB: added new tests, sync clean-cpc-db v1.1
- DB: sync clean-cpc-db v1.0
- DB: documented correctly RMP files
- AUTORUN: added better code inspired by Fredouille – caprice-forever
- DB: new games from retroachievements
- UI: added DB icon DSK to status bar
- DB: you could add direct tokens using $ (for joystick keybinds or cleans)
- DB: added DB v1 using clean-cpc-db info

- CORE: added model 664 to allow DSK and BASIC 1.0
- CORE: detect some configurations from filename
- VIDEO: minor fixes (requires more work)

2048

- Keep big font size on tiles. Keep the font as big as it was for 1 digit all the time.

Dolphin

- Change some logging levels to debug to avoid huge spam in logs
- Add left stick options for IR mode
- Disable Android build
- Expose “Prefetch Custom Textures” core option
- Expose the “Speed Up Disc Transfer Rate” option
- Add anti-aliasing option
- Expose “Immediate xfb” setting”
- Show proper labels when GC controller is used on ports 1-4 in Wii mode
- Fix Kirby’s Air Ride – BlendingState: Set source and destination alpha factors in logic op workaround

PCSX ReARMed

- [LLE] Fixes Black screen when starting a game with Shadow Tower – cdrom: adjust a timing hack
- gpulib: allow commands to span list entries. Fixes bad input delays/no input registered in certain games.
- cdrom: forget old sector on cd change – fixes Syphon Filter 2 does not change disk
- dma: add optional slow linked list walking – fixes issues in Crash Bash (PAL), Spot Goes To Hollywood, Final Fantasy Chronicles (FF4)
- cdrom: make the timing hack conditional – fixes incorrect behavior in Crash Bandicoot
- cdriso: unbreak cdda for chd – fixes sound is broken in several games
- cdrom: don’t report read too early – fixes no (more) background music in Gran Turismo
- lightrec: Disable threaded compiler by default
- spu: fix a wrong assumption – fixes issue in Need for Speed III – Hot Pursuit
- cdriso: fix up chd for separated subq reads
- cdriso: disable the async code. It will race with the new ISOgetBufferSub()
- cdrom: change GetlocP handling – fixes Captain Commando graphical glitches
- cdrom: handle fifo overreads – fixes Dance Dance Revolution 3rd Mix stuck on “NOW LOADING”
- lightrec: Update to the latest Lightrec. Fixes bugs affecting Jackie Chan Stuntmaster, Vib-Ribbon, GTA2, and maybe other games.
- cdrom: add a timing hack – fixes T’ai Fu: Wrath of the Tiger gets stuck on initial loading screen
- cdrom: report read errors correctly
- psxbios: use noninvasive print for -psxout
- adjust multitap code to interact with standalone better
- gpulib: update gpuinfo
- lightrec: fix pcsx interpreter -> lightrec switch too
- lightrec: allow to switch to pcsx’s interpreter anytime
- libretro: adjust options after recent changes
- psxinterpreter: use cycle_multiplier also, not just ari64 dynarec
- remove all the hack options. They just confuse users. For Parasite Eve 2 and others, adjust “PSX CPU clock” instead if needed.
- cdrom: update status immediately after seek. Philosoma wants it
- cdrom: ignore repeated read cmds. Fixes crackling sound in Intro fmv of Suikoden II and similar sound issues, with other games
- cdrom: change GetlocL behavior – fixes Beat Mania (Japan) Notes don’t drop down
- cdrom: treat seek the same as other commands
- cdrom: allow to interrupt initial scan sequence
- cdrom: actually reject commands when not ready. Before it would still execute them and just do an error response
- maybe fix x86 32bit detection for libretro’s infra
- Enable gpu_neon on 32bit x86
- gpu_neon: rm wrong cost qualifier
- cdrom: adjust timing
- don’t cast between long and pointers for win64. long is 32 bit there
- enable gpu_neon on win64 too
- misc: patch some issues indicated by clang
- gpu_neon: brand new x86 SSE2+ implementation
- gpu_neon: adjust some comments and things
- cdrom: delay the missed irq more
- some big endian fixes
- spu: fix wrong volume shift
- psxinterpreter: reduce the use of globals. They induce penalties with -fPIC
- cdrom: partially emulate the fifo
- dma: try more accurate timings. Seems to help Legend of Mana
- cdrom: get rid of pTransfer. It could easily crash the emu (misbehaving game or even malicious ISO)
- cdrom: don’t reschedule irqs. Trigger shortly after ACK instead
- cdrom: clean up command handling
- cdrom: try to eliminate playback timing drifting
- psxcounters: try to eliminate another source of audio drift
- spu: cleanup some irq hacks. Note that bIgnoreLoop is still needed or
  “Misadventures Of Tron Bonne” may hang after cutscenes. Before this commit the game will sometimes cut off dialogues.
- spu: sync on xa playback start. This avoids nasty underflows at the start of the stream
- spu: support master volume
- spu: rename dwChannelOn to something more suitable
- spu: remove some strange rounding. Seems wrong. Also deal with now possible div by 0 in scan_for_irq().

- cdrom: remove another hack. The hack just makes Worms Pinball unstable, and Crusaders of Might
  and Magic no longer seems to need it.
- cdrom: get rid of cdrPlayInterrupt. Unify with cdrReadInterrupt and name it cdrPlaySeekReadInterrupt, although these are not actually interrupts, more like events, but keep this weird PCSX's naming convention
- cdrom: should use the last Setloc loc and SeekL – should fix Ps1 audio slowdown, stutter and slowdown eg tomba 2
- spu: get rid of iXAPitch. Ut makes no sense after SPU was fully synced to the core, and now it
  just pulls unwanted windows.h dependency on win32.
- gpu_neon: fix another abi violation. Before the flush_render_block_buffer call the stack is misaligned, so push an odd num of regs to realign
- gpu_neon: fix wrong block counting
- drc: fix a silly mistake of overwriting a reg
- drc: Rework SMC checks again. The way it was done before wasn't good enough for
  Mega Man Legends 2 at least
- drc: fix wrong masking in set_jump_target
- cspace: add forgotten length decrement
- Switch apple platforms to neon gpu too
- Android – Enable neon gpu for arm64 Android too
- switch most arm64 platforms to neon gpu
- cspace: generic implementation with vector extensions
- gpu_neon: integration for arm64
- gpu_neon: new intrinsics-only implementation. ~80-95% performance of the asm version on cortex-a72,
  but maybe less of a portability nightmare (+arm64 support)
- gpu_neon: place asm func prototypes into a separate header for the upcoming intrinsics implementation
- gpu_neon: don't include vector_ops.h in the main header. That stuff is only used in the C-only prototype
- gpu_neon: fix some more abi violations
- drc: patch up some potential issues on the switch
- gpu_neon: try to make the compiler save some callee-save regs which the asm isn't doing properly
- dfxvideo: patch up some unsafe macros. (old?)-ARM version of GETLE32() was causing wrong shifting due to implicit signed int promotion.
- mman: align and clear for 3ds also callers expect cleared mem
- drc: some more libnx support
- libnx: make mmap act more like on Linux
- lightrec: Fix unmap size of BIOS.
  The BIOS was mapped as 2 MiB, since we want to use a huge page if
  possible. Therefore it should also be unmapped as 2 MiB, otherwise the
  upper 1.5 MiB will still be mapped after de-init, which will make it
  impossible to map the BIOS once again.
- psxmem: Use Lightrec memory map even for interpreter. The CPU emulation method (interpreter or dynarec) can be switched at runtime. Therefore we must use Lightrec's memory map even when the
  interpreter is used.
- frontend: Remove duplicated code. The same function bgr888_to_rgb565() was present twice, protected by different macros.
- Fix PCSX core and dfxvideo for big-endian systems
- Rumble should only work with 'dualshock' device type
- cdriso: hack .chd track handling to match .cue
- cdrom: don't read and play simultaneously. Fixes Sled Storm – Music playing way too fast
- cdrom: Fix PBP support on big-endian platforms
  The data contained in the PBP is in little-endian format. Therefore,
  everything in the PBP's header must be read with the byte-swap macros,
  for PBP support to work on big-endian systems.

- dfxvideo: Restore support for big-endian arch
  The hardware registers, emulated RAM or emulated VRAM are all in
  little-endian format. Therefore, each access must be done with one of
  the byte-swap macros, for the plugin to work on big-endian systems.

- frontend: Fix colorspace conversion routines on big-endian
  The bgr555_to_rgb565() and bgr888_to_rgb565() functions were only
  working correctly on little-endian systems.

- misc: Use GCC builtins for byte-swap operations
  Instead of using custom code to byte-swap values, use the built-in
  function provided by GCC.

- gpulib: Add proper support for big-endian
  Update the gpulib code to work properly on big-endian architectures.

- Get rid of bit fields in union
  Long story short, bit fields aren't endian-safe.
  More info: http://mjfrazer.org/mjfrazer/bitfields/
  Simplify that by just using a few macros to access the needed bits.
- Make sure hardware registers are manipulated as little-endian
  The hardware registers should be represented in little-endian format.
  Therefore, on big-endian systems the values need to be byte-swapped.

- Fix PCSX on big-endian systems
  The __BIGENDIAN__ macro was never defined anywhere, and the
  __BIG_ENDIAN__ macro isn't set anymore by recent versions of GCC.

  Replace them by checking __BYTE_ORDER__ against __ORDER_BIG_ENDIAN__.

- lightrec: Enable code buffer support

Map a 8 MiB code buffer at (base + 0x80.0000), right after the emulated
RAM. In this code buffer, Lightrec will write the recompiled code for
the host machine.

In general, the code buffer support is very useful when the host
platform has only a small RW+X memory area available for JIT purposes,
like it's the case on the WiiU.

On Linux, this isn't a concern, but having a code buffer does still
bring a benefit: if both the start address and the end address of the
code buffer's address fit in 32 bits, then Lightrec's code LUT can be
shrunk in half (2.5 MiB instead of 5 MiB), as it only needs to store
32-bit pointers.

- psxcounters: try to support a dynarec with a very long timeslice
  The dynarec instead should probably not run for thousands of cycles
  doing no interrupt checks, but maybe this hack will be enough. Fixes Resident Evil: DC (DS Ver.) – Memory card is not inserted
- drc: update some logging and patches
- psxmem: Add support for Lightrec's custom mem init sequence
  Adapt the current psxmem.c code to support Lightrec's custom memory
  init functions.
-
  lightrec: Add new memory init functions
  Lightrec can greatly benefit from having the RAM/BIOS/scratchpad and RAM
  mirrors mapped to specific addresses. For the same value of (offset):
  – If the RAM is mapped to (offset), the BIOS is mapped to
  (offset + 0x1fc00000) and the scratchpad to (offset + 0x1f800000), the
  generated code will be better;
  – If the RAM is also mirrored to (offset + 0x200000), (offset + 0x400000)
  and (offset + 0x600000), the generated code will be even better;
  – If the offset is 0x0, the generated code will be even better.

  Additionally, the new memory init code will attempt to use huge pages
  when possible, in order to greatly reduce the overhead of the MMU.
- Add support for mapping emulated RAM to address 0x0
  This requires a few changes, since a pointer whose value is 0x0 will be
  detected as a NULL pointer. The read/write LUTs are now initialized with
  0xff, and all pointers are now checked against a new INVALID_PTR macro.

  Mapping the emulated RAM to the address 0x0 will allow Lightrec to
  generate much better code.
- Update core options to v2 format

Mupen64Plus Next

- Use Fiber backend on Win32.
  Works around driver crashes on AMD Windows since driver seems to rely on unwinds to work properly.
  libco cannot support this without using a more robust backend.
- Update ParaLLEl RSP
- Update ParaLLEl RDP
- ParaLLEl RDP: Workaround Nvidia driver bug on 525.x series with PRIME.
- Update HLE RSP
- Update GlideN64
- update Makefile to enable GLES3 support for rpi4
- Add workaround for IOS/emscripten NPOT Textures mitigation.
- Add iOS AArch64 core

Tyrquake

- Added 384×216 resolution
  Added 384×216 resolution, so we can have a fullscreen, pixel-perfect experience at both 1920×1080 (Full HD) and 3840×2160 (4K).

  Note: for this to work, the aspect ratio "Full" must be set in Settings -> Video -> Scaling.

Beetle Lynx

- Translation updates

Beetle PSX

- Fix eventcycles, add SPU Samples, allows much better performance . EventCycles should work up to 2048 now, now that it is used by MDEC and Timer. SPU samples option was added, audio glitches will occur in some games unless samples is 1
- Core option category adjustments
- Fix 32-bit android buildbot HW builds
- GLES 3.0 Support on HW Renderer

- Fix LED interface crash if not implemented.

Beetle NGP

- Update to 1.29.0
- Set default samplerate to 44KHz and get rid of all other options –
  it is undesirable to run this core at anything above 48KHz if we
  want consistent audio frame pacing
- Get rid of color depth core option – there is no reason to go above 16bit and/or 15bit for NGPC

Beetle PCE

- Add libretro LED interface
- Update to 1.29.0

Beetle PCE Fast

- Update to 1.31.0 –
  * PCE-Fast: Update channel 1 frequency cache upon LFO frequency register writes(the way the channel 1 frequency and LFO frequency are combined
  is still inaccurate, however, causing frequency update timing granularity to be too high).

Beetle Saturn

- CHD: fix block address calculations
  In CHDs tracks are padded to a 4-frame boundary, causing physical and CHD
  block addresses to differ after track 1. Adjusting sector reading to
  account for this offset fixes support of multi-data-track CHDs like Last
  Bronx.
- Android: CHD support enabled
- Fix crash when multitap is enabled
- Backport 1.31.0 changes related to VDP2 –
  SS: When VDP2 per-dot RBG0 coefficient reading is enabled, and a per-dot coefficient read is attempted from a memory region not configured for that purpose
  via RDBS, force the coefficient data to 0, instead of using the per-line coefficient data. Fixes graphical glitches in "Radiant Silvergun" when starting
  a new game after interrupting the attract mode during the AKA-O battle.
  SS: Forcibly disable NBG1-3 only if both RBG0 and RBG1 are enabled via BGON, instead of only checking if RBG1 is enabled; fixes missing/glitchy graphics in
  "Houkago Ren'ai Club: Koi no Etude".
- Backport 1.31.0 database changes, plus take out gettext_noop –
  SS: Added "Real Bout Garou Densetsu" to the internal database of games to enable full cache emulation with, to fix startup hang after skipping the BIOS bootup
  animation, a regression introduced in 1.26.0-UNSTABLE.

Beetle VB

- Update to 1.31.0 –
  * VB: Don't flip framebuffers at the first start of drawing to the
  framebuffer after power-on, and (always) flip framebuffers when XPRST
  is triggered, per tip from blitter

ProSystem

- Abstract MARIA's memory reads into a single function.
  This is needed because SOUPER cartridges can trap MARIA's reads and
  handle them specially.
- Add support for the SOUPER cartridge type, used by "Rikki & Vikki"
- Import the BupBoop library, needed for "Rikki & Vikki".
  This library is needed to emulate the sound hardware that "Rikki &
  Vikki" uses. It's an entirely separate chip included in the cartridge
  that produces 16-bit stereo audio.
  Note that BupBoop supplies a library called WinTone that provides
  DirectSound bindings, but we don't use it both because we already use
  DirectSound and because using WinTone would complicate `libretro`
  portability.
- Add support for the BupChip, used by "Rikki & Vikki".
  Note that this commit itself doesn't actually allow the BupChip to be
  used for anything, because the only way to supply music for the BupChip
  to play is through the CDF format.
- Add support for the BupSystem CDF file format.
  This format allows song files to be supplied to the emulated BupChip for
  playback. It matches the format used by BupSystem:
  http://tailchao.com/BupSystem/index.php
  "Rikki & Vikki" is supplied as a CDF file, along with an unheadered
  `.bin` image and several music files.
- Specify that "Rikki & Vikki" uses the new SOUPER cartridge type in the database.
- Add complete support for "Rikki & Vikki"
- Stop checking size when loading save states.
  The size variable is always zero, because `libretro` saves don't store
  their size explicitly.
  This fixes save states for RAM supercarts.
- Bump the savestate size up for SOUPER carts.
  This makes save states work on "Rikki & Vikki".

- Bump BupBoop music and patch stack depths to 16.
  The author of BupBoop (and “Rikki & Vikki”) let me know that this change
  is necessary for the game music to play properly.
- When reading BupChip file paths in CDFs, replace Windows path separators with Unix ones on non-Windows systems.
  This fixes the Steam version of “Rikki & Vikki” on Unix systems like the
  Steam Deck.
- Fix savestates and music for “Rikki & Vikki”

81

- Change Border Option – Changed the hide border option to add a new border size.
  Previously the full border was displayed or nothing was displayed.
  Now you have three possible options – normal, small, none
- Fix disabled Chrome – Does not allow disabling Chroma 81. The disable option for the “Emulate Chroma 81” option now works

FCEUmm

- Add Firebrandx palettes
- palette.c: Remove unused NTSC palette generator
- m235: Update support for 150-in-1 and 260-in-1 with Contra carts
- Add mapper 441
- Add mapper 463
- Add mapper 466
- Add mapper 464
- Add mapper 459
- Add mapper 465
- Add mapper 461
- Add mapper 460
- Mapper 429: Allow more than 32 KiB CHR
- Add mapper 449
- Add mapper 455
- Mapper 351: Correct CHR mask, add FDS-like mirroring control
- Add mappers 441,449,455,459,460,461,463,464,465,466. Increase maximum CHR size on mapper 429.
- Mapper 449: Add DIP switch/solder pad functionality
- Mapper 460: Add alternative DIP switch/solder pad functionality
- Mapper 351: Add NROM-64 mode
- Mapper 452: Add support for 350-in-1.
- Mapper 268: Rewrite and add submapper variants.
- Mapper 268: Add GenMMC3Close() to Mapper268_close().
- Mapper 268: Correct GNROM mask for submapper 4, correctly apply CHR mask bits in register 2.
- Mapper 432: Add 1 MiB variant
- Add mapper 52 submapper 13 (CHR-ROM+CHR-RAM)
- Mapper 268: correct comment style
- Mapper 40: Add multicart variant
- Add sequential targets light gun support
  Support for Sequential targets Light Guns has been added. “Gun Aux A” serves as light sensor logic input.
- Update zapper.c
  After testing trigger logic invertion option in VS roms, realized that ZD[w].bogo has to be inverted to work properly
- Mapper 40: Add a multicart variant.
- Mapper 52: Add a mixed CHR-ROM+CHR-RAM variant (submapper 13).
- Mapper 268: Rewrite for clarity and add more wiring variants, indicated by submappers. Support mixed CHR-ROM+CHR-RAM boards.
- Mapper 351: Add a rarely-used NROM-64 Mode (NROM with 8 KiB PRG bank size).
- Mappers 432/449/452/460: Minor updates.
- Mapper 268: Write to WRAM before corrupting V when index==2.
- Remove CopyFami stuff
  unused in this implementation, not any other emulator uses it. Considering
  it’s even UNIF based, probably has no use anymore.
- FDS: Permanently use new disk handling functions
  FDS was already using the new disk handling code. This PR just removes
  the old and now unused one.

Mesen

- Made compiling with LTO function (Link-Time Optimization) like in other cores
- UNIF: Add BTL-831128C board (Mapper 528)
- UNIF: Add BTL-900218 board (Mapper 524)
- UNIF: Add BMC-TH2131-1 board (Mapper 308)
- UNIF: Add BMC-TJ-03 board (Mapper 341)
- Mapper 200: Fixed mirroring
- UNIF: Add BMC-SA005-A board (Mapper 338)
- UNIF: Add BMC-L6IN1 board (Mapper 345)
- UNIF: Add BMC-K-3036 board (Mapper 340)
- UNIF: Add BMC-K-3033 board (Mapper 322)
- UNIF: Add BMC-K-3006 board (Mapper 339)
- Mapper 283: Handle both GS-2013/GS-2004 roms in the same mapper
- UNIF: Add BMC-GN-26 board (Mapper 344)
- UNIF: Add BMC-CTC-09 board (Mapper 335)
- UNIF: Add BMC-830134C board (Mapper 315)
- UNIF: Add BMC-10-24-C-A1 board (Mapper 327)

- UNIF: Added support for BMC-S-2009 board (Mapper 434)
- UNIF: Assign some boards to existing ines mappers, Set UNL-8237 as 215.1, Replace Mapper 27 with UNIF CC-21
- Mapper 319 (Eh8813): Fixed latch write registers
- JY Company: Fixed PRG outer banking
- Mapper 15: Fix CHR write protect for some subor/waixing boards
- Implemented input bitmasks

LRMAME

- Update to version 0.250
- Add partial rewind support
- Add partial runahead support
- Auto save/load should work now
- Savestate support should work even for some games that don't support savestates on mainline MAME
- Fix the invalid joystick map message that appeared when using Q-Bert
- Adds Street Fighter Alpha/Zero – CPS Changer to the profile button list
- Set frame timing based off m_frame_period. Upstream derives frame timing from the m_frame_period value in screen.h and not m_refresh as is done in the libretro port. Also adding a check to see if this value has been updated frame to frame to allow the frontend to update timings.
- Clean up lightgun offscreen logic
- Fix mouse buttons, add players 7–8, add multimouse/lightgun
- Add input descriptors for up to 6 players

LRMAME 2003

- Set audio buffer status callback
- Hiscore: It is only copied to memory after successful reading, preventing the use of defined data. For example:
  invadpt2, generate a 0-byte invadpt2.hi, run the emulator at this time, the position of hiscore will display garbled characters
- System 16: Wrestle war ring rotation and titlescreen graphics
- Expand CPU clock scale: add 200%, 250%, 300%.
- BIOS fallback support

LRMAME 2003 Plus

See changelog here (https://github.com/libretro/mame2003-plus-libretro/blob/master/CHANGELOG.md).

LRMAME 2010

- Input system restructure

TGB Dual

- Implement persistent ROM buffer support
- Makefile fix to enable build for "unix-armv7-hardfloat-neon"

Gambatte

- Set OSX target back to 'default' – some libc++ conflicts – rely instead for 10.7 backwards compatibility on us not using anything unusual in libc++ that could cause issues at runtime, like strlcpy_chk_ or sincos
- Update palettes
- Updated 5 palettes + 1 palette name.
  The Game Boy palettes that have been received an update are: Super Saiyan God, Super Saiyan Blue, Super Saiyan, Super Saiyan 3, and AKB48 Pink. Plus, Pepsi Blue from here on out is called Pepsi-Cola Blue, to avoid name confusion with the actual Pepsi Blue soft drink.
- Backport ' Added support for Sonic 3D Blast 5' – from bardeci
- Palette – TWB64 104 – SideM Green Tweaked. The SideM Green Game Boy palette now represents that to the official SideM color as seen on THE IDOLM@STER's official website. It turns out the SideM Green palette was in the incorrect shade all this time until now.
- TWB64 090 and TWB64 140 Updated! Once again, I had to do some deep digging into color research, but just in the Super Dragon Ball Heroes webpages, but regarding Super Saiyan Blue Evolved, the sprite assets from Dragon Ball Z: Dokkan Battle.

NeoCD

- Preliminary support for direct CD-ROM access
- Enabled LTO for some platforms (experimental)
- Reduce latency by adjusting emulated frame slice boundary.
- Add B+C input macro

Picodrive

- core, fix cpu sync regression
- pico, fix psg sound and fast forward
- 32x, minor poll detection fix
- mcd, minor optimization for poll detection
- core, improve mcd+32x cpu synchronization
- 32x, make 68k poll detection less agressive
- sh2 drc (for 32X), fix bug in jump patching for arm64
- core, TH input latency only if output was low
- core, add TH pad pin latency (load state regression)
- core, add TH pad pin latency (regression after 5 min)

- core, add TH pad pin latency after switching to input
- sms, fix center tms text mode (mode 1)
- sms, add fast renderer, remove 1st column (8 px) if blanked
- sms, add xor 4x8k mapper
- mcd, fix gfx (32×32 stamp)
- core+platforms, harmonise supported extensions
- sound, fix ym2413 fm sound load/save
- z80, fix drz80 save (regression)
- core vdp, fix out-of-bounds vram access for save/load
- sms, fix mappers (save/load for sega, korea, 8kbram)
- z80, improve cz80+drz80 compatibility
- z80: fix AF pop+push in DrZ80 (F bits 3+5 stack value unchanged)

ECWolf

- Automap fixes + add “Show Status” button
  There was an incorrect bool preventing 2 options to work properly (automap/overlay rotation and cycle automap/overlay).
  Also changed the inputs while the automap is opened, so the player can still move with overlay or if the normal automap is opened but the pause option is OFF.
  Changed 2 default values, show walls on normal automap and pause while normal automap is open are ON by default now.
  Also added “Show Status” on the unused retropad-Y button, very useful when playing with the screen size set to “Without statusbar”.
- Several memory usage improvements. Mainly meant for RS90 but improving other targets as well
- Decrease internal resolution to 240×160 on RS90
- Disable adlib on rs90
- Synthesize speaker on the fly. It’s faster and uses less memory
- Add proxying semantics in order to avoid loading the same lump for different sounds
- Limit size of loaded audio samples

PX68K

- Crashes fixed at startup
- Fix SIGSEGV (Address boundary error) with Etoile Princess

VBA-M

- Update to core options v2 interface
- HLE: Fix sign-compare warnings
- Fix missing-field-initializer warnings
- Silence implicit-fallthrough warnings
  Used [[fallthrough]]; to mark affected areas. Such keyword should be
  available since C++11

- Wrap functions unused by LIBRETRO
- Simplify save state handling
  – size and version parameters are unused in the libretro implementation,
  so remove them
- Cleanup more libretro-specific branches

Mu

- Add build for Mac OS ARM.
- Update valid extensions to everything in the core info; Disallow libretro from looking into ZIP files, we will be handling it ourself so that we can load all PRCs and PDBs.
- Include miniz library; Make all CMake builds PIC.
- RetroArch CMake corrections; Implement loading from ZIP files which contain PRCs and PDBs.

NOTE: To load multiple files at once, place PRCs and PDBs into a ZIP file and then
load that ZIP file. PDBs will be installed first followed by PRC files.

Desmume

- Update core options v2 interface
- Hide CPU mode option when HAVE_JIT is 0 & OpenGL exclusive options when using soft3D

Genesis Plus GX

- Update to the latest version
- Enchanced per-tile vertical scroll implementation

Genesis Plus GX Wide

- Update to the latest version
- Enchanced per-tile vertical scroll implementation

NOTE: Enhanced per-tile vertical scroll – This emulation hack allows allows each cell to be vscrolled individually, instead of being limited to 2-cell (16px). The offset of the new, intermediary cell is calculated as an average of the offset of the current 2-cell and the offset of the next 2-cell.

See this thread here (https://github.com/libretro/Genesis-Plus-GX/pull/306) for before/after comparisons.

Citra

- Fix pointer being hidden on single screen mode
- Expose CPU clock scaling option
- First minimal implementation of Libretro VFS
- Avoid caching framebuffer in core. Frontend should be able to change it at any time.
- Android port/build

SameBoy

- Update core with latest changes (August 2022)

Boom3

- Makefile fix to enable build for unix-* platforms

SameCDi

- Save NVRAM on per directory basis
  CDI NVRAM is extremely small, so saving to it becomes an issue when you have many games. This separates NVRAM into game specific dirs

- Add Redump BIN/CUE format
  Hack to support CDI/2352 format, per hack listed [here](https://github.com/mamedev/mame/pull/8529/files)

- CHD support

Atari800

- OSX 10.7 backwards compatibility
- Fix PSVita build

gpSP

- Fix Pokemon Fire Red germany not booting/ White Screen
  Wrong Gamepak Code causing Pokemon Fire Red not to boot due to wrong flash size (64k instead of 128k)

RACE

- OSX 10.7 backwards compatibility

Opera

- Lightgun – allow offscreen reloading on all 4 sides of the screen –
  Currently, many 3DO games only allow lightgun users to shoot off parts of the top of the screen in order to reload. This code change allows lightgun users to fire off any part of any side of the screen to reload in games that have a functioning reload/holster input. It is tested as working with all 3DO lightgun games except for Mad Dog McCree and Space Pirates, which still exhibit the issues outlined in #176, and it is also tested as working with the arcade game Shootout at Old Tucson (which didn't appear to have functioning offscreen reload before this).

Beetle PSX (https://www.libretro.com/index.php/category/beetle-psx/), Beetle Saturn (https://www.libretro.com/index.php/category/beetle-saturn/), Blog (https://www.libretro.com/index.php/category/blog/), Citra (https://www.libretro.com/index.php/category/citra/), ECWolf (https://www.libretro.com/index.php/category/ecwolf/), FBNeo (https://www.libretro.com/index.php/category/fbneo/), Genesis Plus GX (https://www.libretro.com/index.php/category/genesis-plus-gx/), Genesis Plus GX Wide (https://www.libretro.com/index.php/category/genesis-plus-gx-wide/), Mesen (https://www.libretro.com/index.php/category/mesen/), Mesen S (https://www.libretro.com/index.php/category/mesen-s/), mGBA (https://www.libretro.com/index.php/category/mgba/), Mu (https://www.libretro.com/index.php/category/mu/), Picodrive (https://www.libretro.com/index.php/category/picodrive/), Same_CDi (https://www.libretro.com/index.php/category/same_cdi/), SameBoy (https://www.libretro.com/index.php/category/sameboy/)

RetroArch 1.13.0 release (https://www.libretro.com/index.php/retroarch-1-13-0-release/)

November 20, 2022 (https://www.libretro.com/index.php/retroarch-1-13-0-release/)

RetroArch 1.13.0 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

NOTE: Since it has been a very long time since the last core progress update report, there will be a Core Progress report very soon listing all the changes over the past 6 months that have been made to all the cores in our repertoire. A ton of stuff has happened, lots of things have been improved, and it'd be criminally negligient not to talk about it at all. So stay tuned for that blog post.

Android

The feature 'Vibrate On Key Press' has been enabled by default now on Android. You will get haptic feedback when pressing any of the overlay gamepad elements onscreen, improving the user experience.

In addition, several big improvements are being made under the hood to improve and refine overlay touchscreen controls. Read more about that later in this article (see 'Overlay / onscreen gamepad controls improvements').

Threaded Video is also disabled now by default on new installs/configs. We have left this setting on since 2013 when frame time spikes were dreadful on Android due to SoCs being underpowered and tons of processes running in the background. Fastforward nearly 10 years later and we feel times have changed since and it's a little safer to disable it now. Threaded video can theoretically be faster than non-threaded video but also leads to more judder and less precise frame times. Non-threaded video therefore (if frame time deviation is below 5% margins) should lead to smoother frame times. It's still possible for the user to enable Threaded video if they desire, so nothing has been lost in the process.

Note – another benefit of non-threaded video being the default is that it fixes some issues that were experienced upon focus loss of an app and re-entering RetroArch.

iOS 13+ port

Thanks to the new Swift backend targeting iOS 13 and up, it has become easier to add several new QoL features, such as iPad trackpad support for iOS 13.4 and above.

macOS

Users upgrading to the recent macOS Ventura would have discovered that RetroArch had issues going into fullscreen. This has been fixed in this newer version.

Some keyboard keys have also been hooked up that were not added to the key symbol map before, such as Z/X, and the left meta key.

Overlay / onscreen gamepad controls improvements

Android/iOS users are going to benefit a lot in this release and upcoming versions from various improvements being made to neil4 to overhaul and improve touch controls.

Several gamepad overlays have already been updated to take advantage of these new features. These are as follows:

- gamepads/neoretropad
- gamepads/neoretropad-clear

There are also new 'lite' overlays that take advantage of all the new features described below. In particular, the lite overlays rely heavily on 'exclusive' hitboxes.

Let's go into what has been added so far to this release:

Add eightway area types

- New descriptors: 'dpad_area' and 'abxy_area'

- Each has a diagonal sensitivity setting, 100% being 8-way symmetry
- Buttons can be redefined in the cfg file

E.g. This would create a d-pad area, then redefine it to use analog directions:

```
overlay0_desc0 = “dpad_area,0.85,0.57,rect,0.166228,0.295516”
overlay0_desc0_up = “r_y_minus”
overlay0_desc0_down = “r_y_plus”
overlay0_desc0_left = “r_x_minus”
overlay0_desc0_right = “r_x_plus”
```

Note: If ‘Hide Overlay in Menu’ is enabled and the menu is up, the overlay preset will be read (loaded & unloaded without being shown) to know which overlay types are in the preset, i.e. whether to show eightway settings or not.

Add ‘reach’ and ‘exclusive’ for hitboxes

These allow stretching (or shrinking) hitboxes and handling their overlap. Does not affect image, analog range, or analog/eightway center.

Hitbox Reach

reach_up, reach_down, reach_left, reach_right:

- Stretches hitbox in one direction:

reach_x, reach_y

- Stretches hitbox symmetrically

E.g. In the overlay cfg, this creates a D-Pad area and extends its hitbox left & right 50%, up 15%, and down 30%:

```
overlay0_desc0 = “dpad_area,0.15,0.57,rect,0.166228,0.295516”
overlay0_desc0_reach_x = 1.5
overlay0_desc0_reach_up = 1.15
overlay0_desc0_reach_down = 1.3
```

Exclusive Hitboxes

exclusive:

If true, blocks input from overlapped hitboxes

range_mod_exclusive:

Similar, but only applies when this hitbox is extended by range_mod
After range_mod takes effect, has priority over ‘exclusive’
E.g.

```
overlay0_desc0_exclusive = true
overlay0_desc1_range_mod_exclusive = true
```

Expected use cases:

- A ‘range_mod_exclusive’ hitbox placed close enough to other hitboxes that its range_mod value would otherwise cause unwanted overlap.
- A smaller ‘exclusive’ hitbox placed inside (or partially inside) a larger one; the smaller hitbox would “carve out” space for itself.
- A combination of the two: A smaller ‘exclusive’ hitbox carves out space for itself, but the larger hitbox has ‘range_mod_exclusive’. Whichever is hit first effectively has priority.

Ignore hitboxes with zero area

I.e. Set ‘reach_x’ or ‘reach_y’ to zero to ensure no hitbox math is done.

This simplifies designating animation-only descriptors (e.g. for eightway areas) or obsolete descriptors.

Content scanning

PS1 and PS2 content scanning has been improved significantly in RetroArch. All PS2 discs should now be able to be scanned. Previously, only CD-based PS2 games could be scanned and not DVD-based ones.

PS1 content scanning has also been improved. More content should be able to be recognized now that the system is also able to scan PSX.EXE files on a disc. For instance, it was not possible to scan ‘Street Fighter: Real Battle on Film’ before because of this. Also, all LSP- titles were previously ignored, which has also been fixed.

Menu QoL improvements

Various improvements made to the menu system in terms of usability and UX thanks to sonninos.

- Quick Menu entries were pretty random in regards of doing scrolling with left & right, so it's been cleaned up a bit, and we made sure all submenus behave in unison.

- Ozone – Show metadata helper in footer only with second thumbnail. Removed metadata toggle helper from footer when second thumbnail is not enabled – which is the default – because the button won't do anything without second thumbnail enabled.
- We made the background image selector usage more comfortable by starting from assets directory or current selection instead of drive letter root, and showing image previews while browsing.
- File selector globally removes file from selection path, uses path instead and sets file as selection
- File selector title corrected to have : separator
- It's possible to control the menu now with the right analog stick, or both at the same time if you want to do so.
- Cursor memory for most menu screens
- (Applicable to Ozone) Footer improvements – show metadata helper in footer only with second thumbnail, add "Cycle thumbnails" helper when suitable, show "Search" helper only when search function is enabled, fix "Thumbnails available" helper for save states, tighten padding between icon and title, and widen between helpers

For a more exhaustive list of all changes made to the menu, consult the Changelog.

Vulkan fix for HDR mode

After RetroArch 1.11.1, an unexpected side effect caused many shaders (e.g. crt-easymode-halation or newpixie.crt) to render an oversaturated and dark image when using Vulkan and HDR on Windows 10/11. To fix this now with the Vulkan driver, we only skip the tonemapper if HDR10 is explicitly enabled by the last shader pass. Otherwise, we are simply just inheriting the bit-depth of the swapchain.

Changelog

1.13.0

- 3DS: Remove debug button combo to shutdown RA
- 3DS: Remove MaterialUI as per MrHuu recommendation
- ANDROID: Enable 'Vibrate On Key Press' haptics by default
- ANDROID: Turn 'Threaded Video' off by default
- CHEEVOS: Upgrade to rcheevos 10.5
- COMPILATION: Fixed compiling with –disable-menu
- CONFIG: Don't show override notification with appendconfig alone
- DATABASE/PLAYLISTS: Playlist + database changes – Cleanup 'entry_slot', fallback label + logging
- FRONTEND: Fix default remaps folder for various cores: remap should … …be nested in config folder
- HOTKEYS: Fix shader toggle and add hotkey + sublabel
- HOTKEYS: Cleanups and corrections – Keep hotkey pause and menu pause separate in order to not trigger unwanted pause when toggling menu regardless if menu will pause or not
- HOTKEYS: Cleanups and corrections – Allow unpausing with Start (makes resuming more convenient after controller disconnect if menu does not pause)
- IOS13+: Pointer movement accuracy. iPad Trackpad Pointer Movement Accuracy through absolute location (for iOS 13.4 and above)
- IOS13+: Adds iPad Trackpad Support to iOS13 Project (for iOS 13.4 and above)
- INPUT: Fixed the way devices were previously indexed. Input devices were only being indexed in order and would stop at the first time an input has no device connected to it. The problem is when a device gets disconnected, that input will have no devices connected to it, but the next input may still have a device connected. So, that makes changing the port of the currently connected devices impossible.
- INPUT/AUTOCONFIG: Add option for pause on controller disconnect
- INPUT/AUTOCONFIG: Driver independent disconnection notification. Should show disconnect notification now properly on Windows with XInput and/or DirectInput pads
- INPUT/HID: Added usb hid controllers for the famous ZeroDelay encoder and also for "Kade: Kick Ass Dynamic Encoder" to be able to use some custom arcade sticks.
- INPUT/OVERLAY: Add eightway area types.
- INPUT/OVERLAY: Fix overlay next_index for unnamed targets
- INPUT/OVERLAY: Ignore hitboxes with zero area. I.e. Set 'reach_x' or 'reach_y' to zero to ensure no hitbox math is done. This simplifies designating animation-only descriptors (e.g. for eightway areas) or obsolete descriptors.
- INPUT/OVERLAY: Add 'reach' and 'exclusive' for hitboxes. Allows stretching hitboxes and handling their overlap.
- INPUT/MENU: Addition to analog stick menu navigation
- INPUT/MENU: Enable menu navigation also with right analog stick
- INPUT/MENU: Add option for swapping menu scrolling buttons
- LOCALIZATION: Updates
- LOCALIZATION: Add Hungarian language option
- MENU: Thumbnail fullscreen toggle behavior correction
- MENU: Consistent left-right scrolling for Quick Menu items
- MENU: Remove useless sublabel from System Information
- MENU: Improve widget appearance with missing assets
- MENU/QT/WIMP: Remove SSL/TLS check at startup
- MENU/OZONE: Show metadata helper in footer only with second thumbnail
- MENU/OZONE: Footer improvements – Add "Cycle thumbnails" helper when suitable
- MENU/OZONE: Footer improvements – Show "Search" helper only when search function is enabled
- MENU/OZONE: Footer improvements – Fix "Thumbnails available" helper for save states
- MENU/OZONE: Footer improvements – Tighten padding between icon and title, and widen between helpers
- MENU/OZONE: Launching anything from a View no longer throws Quick Menu off the screen

- MENU/OZONE: Save state thumbnails in slot dropdown obeys fullscreen toggle properly when content launched via CLI
- MENU/OZONE: Save state thumbnail dropdown won't allow fullscreen toggle when it shouldn't
- MENU/OZONE: Selection position remembering in non-playlists won't flash the first entry
- MENU/OZONE: Remember selection per main tabs
- MENU/OZONE: Remove incomplete assets warning
- MENU/OZONE: Add option to adjust cursor memory when changing menu tabs
- MENU/OZONE: Further extend texture support for Core Option categories
- MENU/XMB: Remove incomplete assets warning
- MENU/XMB: Add truncate playlist name option
- MENU/XMB: Improve background image selector
- MENU/XMB: Add option to adjust cursor memory when changing menu tabs
- MENU/XMB: Further extend texture support for Core Option categories
- MENU/MATERIALUI: Remove incomplete assets warning
- OSX: Fixed Z/X keys not working on the macOS port
- OSX: Fixed RETROK_LMETA not working on macOS port. The RETROK_LMETA key was not defined in the rarch_key_map_apple_hid
- OSX: Fix broken fullscreen mode in macOS Ventura
- PS2: Fix Error saving remaps and runtime logs
- PS3: Fix Core Remap Overwrite Fail
- QB: Don't fail if OSDependent/OGLCompiler libraries are not present
- SCANNER/PS1: Improved scanning of PS1 discs
- SCANNER/PS2: Added serial scanning of PS2 discs – should now scan DVDs and other discs which were previously missed
- THREADED VIDEO/GLCORE: Fix regression 'Shader presets dont load, when video driver is set to glcore'
- VULKAN: Fix HDR inverse tonemapping. Only skip tonemapper if HDR10 is explicitly enabled by last shader pass. Otherwise, we are simply just inheriting the bit-depth of the swapchain.

Blog (https://www.libretro.com/index.php/category/blog/)

RetroArch 1.12.0 release (https://www.libretro.com/index.php/retroarch-1-12-0-release/)

October 18, 2022 (https://www.libretro.com/index.php/retroarch-1-12-0-release/)

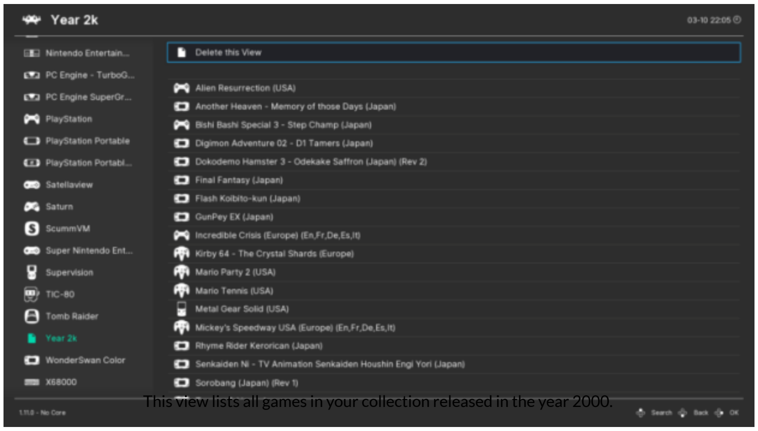
RetroArch 1.12.0 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

Views

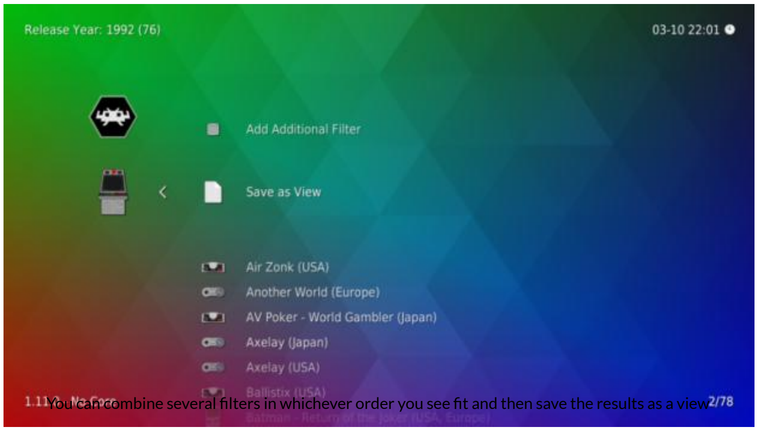

Year 2k
03-10 22:05
Nintendo Entertain...
PC Engine - TurboG...
PC Engine SuperGr...
PlayStation
PlayStation Portable
PlayStation Portabl...
Satellaview
Saturn
ScummVM
Super Nintendo Ent...
Supervision
TIC-80
Tomb Raider
Year 2k
WonderSwan Color
X68000
Delete this View
Alien Resurrection (USA)
Another Heaven - Memory of those Days (Japan)
Bishi Bashi Special 3 - Step Champ (Japan)
Digimon Adventure 02 - D1 Tamers (Japan)
Dokodemo Hamster 3 - Odekake Saffron (Japan) (Rev 2)
Final Fantasy (Japan)
Flash Koibito-kun (Japan)
GunPey EX (Japan)
Incredible Crisis (Europe) (En,Fr,De,Es,It)
Kirby 64 - The Crystal Shards (Europe)
Mario Party 2 (USA)
Mario Tennis (USA)
Metal Gear Solid (USA)
Mickey's Speedway USA (Europe) (En,Fr,De,Es,It)
Rhyme Rider Kerorican (Japan)
Senkaiden Ni - TV Animation Senkaiden Houshin Engi Yori (Japan)
Sorobang (Japan) (Rev 1)
1.11.0 - No Core
Search
Back
OK

This view lists all games in your collection released in the year 2000.

There is now an entirely new way to display and organize content – Views!

You can turn a filter set in the Explore menu into a so called “View” file which then gets listed alongside playlists. This also adds the ability to filter a category by range in the Explore menu and not just filter on exact matches.

The views are saved into .lvw (libretro view) files that just like playlist .lpl (libretro playlist) files are in JSON format and are stored in the same playlists directory.



You can combine several filters in whichever order you see fit and then save the results as a view

Here’s some examples of views that are now possible with the new range filtering:

1. Games made between 1992 and 1996 by developers Capcom and Capcom Production Studio 1
2. Games supporting 5 to 8 players
3. Games containing “Mario” in the title released between 1988 and 1994

These are just some examples of what is possible with this new system. It’s a pretty powerful system, and we hope this gives you a lot of flexibility to better organize your content.

How to use it

1. Go to Explore screen/tab.
2. Search through the list of games sorting through it however you want, then once you’re satisfied with the output, select ‘Save As View’. It will now add this ‘view’ to the playlist list. In Ozone, you can select it from the left popup toolbar. In XMB it will show up next to your playlists. NOTE: You can currently not rename views from within the application.

Android – Haptic feedback for touch overlays

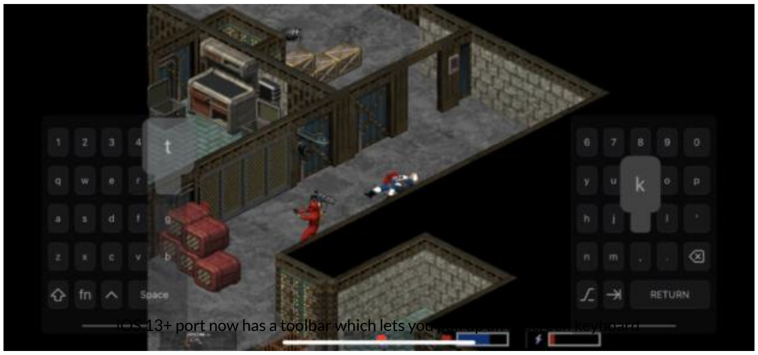
Courtesy of neil4 who implemented this great feature
This repurposes the setting ‘Vibrate On Key Press’, which previously enabled a 16ms vibration on any screen touch.

Now the device’s standard keypress feedback is triggered on overlay key/button state changes, except when a finger leaves the screen.

It’s currently only implemented for Android, but it can potentially be implemented for other systems with haptic feedback later. It’s not enabled by default for now so you will have to enable it manually. To do this, go to **Settings -> Input -> Haptic Feedback/Vibration**, and enable **‘Vibrate On Key Press’**.

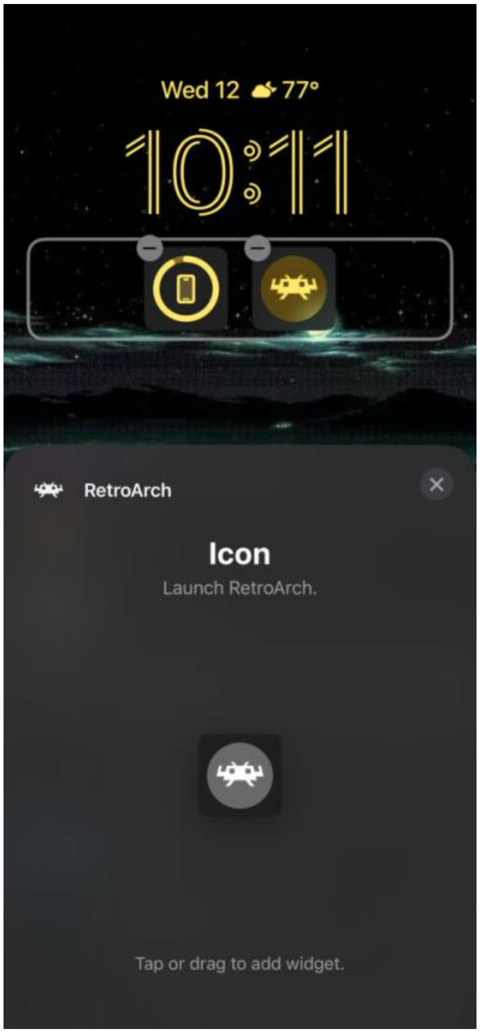
iOS 13+ port

The iOS ARM64 port is completely revamped and targets iOS 13 and later now. It leverages Swift and it has some unique features.



iOS 13+ port now has a toolbar which lets you pull up an onscreen keyboard

For instance, it adds support for revealing the onscreen keyboard and enabling touch screen mouse input by adding a toolbar that is revealed by tapping the top of the screen.



There is also now an iOS 16 lockscreen widget.

WiiU improvements

Some important WiiU platform improvements. Default directories should now be created on the fly (fixing a longstanding issue), and some of the networking issues that popped up in 1.11.x have now been fixed.

Changelog

1.12.0

- CONFIG/CLI: Allow use of –appendconfig with override cfgs instead of getting ignored
- CONFIG/LOGGING: No more console startup logging if logging to file
- CONFIG: Saves config on exit only once instead of (worst case scenario) 3 times
- DATABASE/EXPLORE/VIEW: Add View feature – Add saving of a filter set in the Explore menu into a so called “View” file which then gets listed alongside playlists. This also adds the ability to filter a category by range in the Explore menu and not just filter on exact matches.
- FILEBROWSER: Fix regression – certain extensions keep disappearing

- IOS: New modern iOS version (targeting iOS 13 and up), leverages Swift
- IOS: Fixes to iOS toolbar
- IOS16: Add iOS 16 lock screen widget
- IOS13+: Added emulator keyboard
- IOS13+: Add JIT support for non-jailbroken devices
- IOS13+: Added support for touch mouse handler
- IOS13+: Changed click-and-drag behavior to double tap hold and drag
- INPUT/HAPTIC/OVERLAYS/ANDROID: Improve haptic feedback for input overlays
- LINUX/MALI FBDEV: Add conditional support for OpenGL ES 3.x
- LOCALIZATION: Updates
- LOCALIZATION/ENGLISH: Add British English language option
- LOGGING/QT: Increase log buffer to 2048 characters – Vulkan validation layer messages output correctly now.
- MENU/XMB: Remember selection per main tabs. Addresses the following : collection playlists can contain hundreds or thousands of items. When scrolling through one, pressing left or right by accident can be common. This resets the playlist to the top
- MIST/STEAM/STEAMDECK: Don't expose Black Frame Insertion (BFI) if we are running on a Steam Deck
- NETWORKING/WINDOWS: Disable poll support for MSVC 2010 and earlier. WSAPoll is not supported on Windows XP and earlier.
- NETWORKING/WIIU: Fix socket_connect_with_timeout for WIIU
- NETWORKING/WIIU: Fixes RetroAchievements login
- NETWORKING/WIIU: Fixes other online updater functionality
- SAVESTATES/NOTIFICATIONS: Add delay to savestate notifications, so that GPU savestate screenshots stay untouched
- SAVESTATES/SCREENSHOTS: Avoid 'video_gpu_screenshot' with savestates. Allow GPU screenshots with savestates only when there is no other way of getting a screenshot.
- SCREENSHOTS/VULKAN: Unload screenshot widget texture early. Fixes Vulkan crash when closing content while a screenshot widget is still on-screen
- SCREENSHOTS/VULKAN: Fix screenshot widget crash when ticker animating
- WAYLAND: Set correct app ID
- WIIU: Add some missing default directories
- WIIU: Get mkdir working on WiiU (directory creation)

Blog (https://www.libretro.com/index.php/category/blog/), Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

RetroArch 1.11.1 release! (https://www.libretro.com/index.php/retroarch-1-11-0-release/)

September 30, 2022 (https://www.libretro.com/index.php/retroarch-1-11-0-release/)

RetroArch 1.11.1 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

NOTE: The Android version on Samsung Galaxy Store, Huawei AppGallery, and Amazon App Store will be updated soon. We will remove this notice when it has been updated. Until then, grab the APK from our site.

NOTE: Several size optimizations have been made to the packages. We no longer pre-install all of the optional XMB theme packs or other miscellaneous assets. Previously we also shipped autoconfig files that were irrelevant for that specific platform. By excluding these files from the package, we have managed to reduce the filesize and overall amount of files of RetroArch downloads/installs significantly. On consoles this will be very helpful where SD card/FTP installs can tend to be very slow.

If you still want to have all assets, you can go to Online Updater and select 'Update Assets'. This will install all assets.

Changelog

1.11.1

- GENERAL: Fix DEFAULT_FILL_TITLE_MACRO
- NETWORKING: Add the const qualifier to some function parameters
- NETWORKING/NETPLAY/UPNP: Add a private or CGNAT address warning to UPnP
- SAVESTATES/SCREENSHOTS: Avoid 'video_gpu_screenshot' with savestates
- UWP: Better 'Save on quit' fix

1.11.0

- 3DS: Add unique ID's
- 3DS: Add bottom menu options
- 3DS: Set bottom_asset directory default
- 3DS: Only enable internal counter with CONSOLE_LOG defined
- 3DS: Set default bottom font values
- 3DS: Fix CIA installation issues
- 3DS: Support latest libctru
- ANDROID: Add HAVE_ACCESSIBILITY
- ANDROID: Gingerbread support
- ANDROID: Touchpads support
- ANDROID: Builtin Xperia Play autoconfig profile
- ANDROID: Disable Feral GameMode for Android – only available on Linux
- ANDROID: Add a configurable workaround for Android reconnecting devices
- ANDROID/FDROID: Add F-Droid metadata to repo in Fastlane format
- AUDIO/AUDIO MIXER: Add missing locks for thread safety
- AUDIO/AUDIO MIXER: Fix audio mixer memory leak + remove redundant 'single threaded' rthreads implementation
- AUTOSAVE: Change/improve exit behavior of autosave thread – if condition variable is signaled, the loop is ran another last time so we can do a final check/save before stopping the thread.
- CDROM: Fix memory leak caught with asan – buf passed to filestream_read_file
- CORE INFO/NETPLAY: Ensure current core info is initialized at runloop_event_init_core when netplay is enabled
- CHEEVOS: Upgrade to rcheevos 10.4
- CHEEVOS: Allow creating auto savestate in hardcore
- CHEEVOS: prevent invalid memory reference if game has achievements but core doesn't expose memory
- CHEEVOS: Release achievement badge textures when video driver is deinitialized
- CHEEVOS: Re-enforce hardcore limitations once achievements are loaded
- CHEEVOS/MENU/MATERIALUI: Show achievement badge icons in MaterialUI driver
- D3D9: D3D9 has been split up into two drivers – D3D9 HLSL (max compatibility, no shader support yet) and D3D9 Cg (dependent on deprecated Nvidia Cg runtime library)
- D3D9/HLSL/XMB: XMB fix
- D3D9/CG: D3D9 Cg driver fixed
- D3D11: Fix overlay not showing up
- D3D11/12: Reduce lag with WaitForVBlank – this rather simple addition seems to make D3D11/12 very very close to Vulkan/GLCore regarding input lag.
- D3D11/12: Add waitable swapchains and max frame latency option
- D3D11/12: Make waitable swapchains optional
- DATABASE: Reformat 'rdb_entry_int' – Nitpick adjustments for database entries: Capitalize "Release Date", and remove space before : from Release Date rows which use integer
- DATABASE/EXPLORE: Allow On-Demand Thumbnails in Explore menu
- DATABASE/EXPLORE/MENU/OZONE/XMB/RGUI: Explore menu thumbnails
- DISC CONTROL: Better Disc Control append focus
- DOS/DJGPP: Add a workaround for libc bug
- AUTOMATIC FRAME DELAY: Added slowmotion resiliency
- AUTOMATIC FRAME DELAY: Added string representation for seeing the current effective delay without opening statistics
- AUTOMATIC FRAME DELAY: Added "ms" to logging and "(ms)" to label just like in Audio Latency
- GENERAL: Don't bake in OpenAL and libcaca by default unless explicitly enabled with configure switch.
- GENERAL: Reduce amount of strlen calls
- GENERAL: Reduce or simply sin/cosf calls
- GFX: Fix readability and precision issues in aspectratio_lut
- GFX: Add option to manually enable/disable automatic refresh rate switching
- GFX: Enable automatic configuration of 'VSync Swap Interval'
- GFX/FONT/FREETYPE: Use FT_New_Memory_Face – first read it from file to memory beforehand –
this solves an asset extraction issue when selecting 'Update Assets' – apparently FT_New_Face keeps an open file handle to the font file which
prevents it from being overwritten/deleted while the program is still running.
- GFX/THUMBNAILS: Thumbnail aspect ratio fix
- GFX/THREADED VIDEO: Optimizations, fixes and cleanups
- GFX/VIDEO FILTERS: Add Upscale_240x160-320×240 video filter with 'mixed' method
- GLSLANG: Fix compilation with ./configure –disable-builtinglslang – was missing linking against -lMachineIndependent and -lGenericCodeGen static libs
- INPUT: Fix off by one error for input_block_timeout setting. Also default to 0 for this setting (pretty massive performance gain)
- INPUT: Analog button mapping fixes
- INPUT/HID/OSX: Fix DualShock3 support
- INPUT/HID/LINUX: (qb) Disable HAVE_HID by default for now for Linux as long as there are no working backends for both
- INPUT/HID/WINDOWS: (qb) Disable HAVE_HID by default for now for Windows as long as there are no working backends for both

- INPUT/HID/WIIU: Fix DualShock3 support
- INPUT/OVERLAY: Block pointer input when overlay is pressed
- INPUT/REMAPPING: input_remapping_save_file – existing remapping file was needlessly reloaded
- INPUT/REMAPPING: Add option to disable automatic saving of input remap files
- INPUT/LINUX/UDEV: Fix lightgun scaling on Y axis
- INPUT/LINUX/X11/LED: Add LED keyboard driver
- INPUT/WINDOWS/LED: LED keyboard driver cleanup
- INPUT/WINDOWS/WINRAW: Clear key states when unfocused
- INPUT/WINDOWS/WINRAW: Fix pointer device position
- IOS: iOS app icon fixes & revisions
- LIBRETRO/SAVESTATES: Implement an api call for context awareness
- LOCALIZATION: Updates
- LOCALIZATION: Add Catalan language option
- LOCALIZATION: Fix some bad localization
- LINUX: Make memfd_create call more backwards compatible by calling it through syscall – on older systems, you'll have to include linux/memfd.h for the MFD_ defines, and call memfd_create() via the the syscall(2) wrapper (and include unistd.h and sys/syscall.h for it work). We exclude linux/memfd.h header include because we already provide the MFD_ defines in case they are missing
- LINUX/MALI FBDEV: Fix assertion failed on video threaded switch
- MENU: Menu paging navigation adjustments
- MENU: New Menu Items for disabling Info & Search buttons in the menu
- MENU: Allow the user to use volume up/down/mute hotkeys from within the menu
- MENU: Add missing sublabels for non-running Quick Menu
- MENU: Reorganize Quick Menu Information
- MENU: Savestate thumbnails – Savestate slot reset action
- MENU: Allow changing savestate slots with left/right on save/load
- MENU: Add 'Ago' to playlist last played styles
- MENU: Add proper icons for shader items
- MENU/MATERIALUI: Add icon for 'Download Thumbnails'
- MENU/XMB: Add options for hiding header and horizontal title margin
- MENU/XMB: Dynamic wallpaper fixes
- MENU/XMB: Add Daite XMB Icon Theme
- MENU/XMB/OZONE: Savestate thumbnail aspect ratio
- MENU/XMB/OZONE: Core option category icon refinements
- MENU/XMB/OZONE: Fullscreen thumbnail browsing
- MENU/XMB/OZONE: Add playlist icons under 'Load Content'
- MENU/XMB/OZONE: Thumbnail improvements
- MENU/XMB/OZONE: Savestate thumbnail fullscreen + dropdown
- MENU/XMB/OZONE: Prevent unnecessary thumbnail requests when scrolling through playlists
- MENU/OZONE: Fix playlist thumbnail mouse hover after returning from Quick Menu
- MENU/OZONE: Thumbnail visibility corrections
- MENU/OZONE: Playlist metadata reformat
- MENU/OZONE: Savestate thumbnail fixes
- MENU/OZONE: Add savestate thumbnails
- MENU/OZONE: Header icon spacing adjustment
- MENU/RGUI: Savestate thumbnails
- MENU/SETTINGS: Turn Advanced Settings on by default, this entire filtering of settings will need a complete rethink anyways
- MENU/WIDGETS: Widget color + position adjustments
- MIYOO: Exclude unused HAVE_HID for Miyoo
- MIYOO: Enable screenshots
- MIYOO: Enable rewind
- NETWORK: Allow MITM server selection on OK callback
- NETWORK: Replace socket_select calls
- NETWORK: Implement binary network streams
- NETWORK: Poll support
- NETWORK: Check connect errno for successful connection
- NETWORK: Get rid of the timeout_enable parameter for socket_connect
- NETWORK: Fix getnameinfo_retro's port value for HAVE_SOCKET_LEGACY platforms
- NETWORK: Define inet_ntop and inet_pton for older Windows versions
- NETWORK: Define isinprogress function
- NETWORK/NATT: Move natt files to "network"
- NETWORK/NETWORK STREAMS: Add function netstream_eof
- NETWORK/NETPLAY: Fix game CRC parsing
- NETWORK/NETPLAY: Disable and hide stateless mode
- NETWORK/NETPLAY: Change default for input sharing to "no sharing"
- NETWORK/NETPLAY: Enforce a timeout during connection
- NETWORK/NETPLAY: Disallow clients from loading states and resetting
- NETWORK/NETPLAY: Special saves directory for client
- NETWORK/NETPLAY: Ensure current content is reloaded before joining a host
- NETWORK/NETPLAY: Fix client info devices index
- NETWORK/NETPLAY: Fix input for some cores when hosting
- NETWORK/NETPLAY: Memory leak fixes
- NETWORK/NETPLAY: Force a core update when starting netplay
- NETWORK/NETPLAY: Fix NAT traversal announce for HAVE_SOCKET_LEGACY platforms
- NETWORK/NETPLAY: Refactor fork arguments
- NETWORK/NETPLAY: Fix content reload deadlocks on static core platforms
- NETWORK/NETPLAY: Disallow netplay start when content is not loaded for static core platforms
- NETWORK/NETPLAY: Show client slowdown information
- NETWORK/NETPLAY: Improve check frames menu entry
- NETWORK/NETPLAY: Do not try to receive new data if the data is in the buffer

- NETWORK/NETPLAY: Copy data on receive, even if the buffer is full
- NETWORK/NETPLAY: Fix lobby sublabel CRC display on some platforms
- NETWORK/NETPLAY: Support for customizing chat colors
- NETWORK/NETPLAY: Small launch compatibility patch adjustments
- NETWORK/NETPLAY: Support for banning clients
- NETWORK/NETPLAY: Minor tweaks to the find content task
- NETWORK/NETPLAY: Support for gathering client info and kicking
- NETWORK/NETPLAY: Fix possible deadlock
- NETWORK/NETPLAY: Initialize client's allow_pausing to true
- NETWORK/NETPLAY: Disable netplay for unsupported cores – with stateless mode being disabled for now, there is no reason not to include this. Refuse to initialize netplay when the current core is not supported (no proper savestates support)
- NETWORK/NETPLAY/DISCOVERY: Ensure fixed width ints on packet struct
- NETWORK/NETPLAY/DISCOVERY: Support for IPv4 tunneling (6to4)
- NETWORK/NETPLAY/DISCOVERY/TASKS: Netplay/LAN Discovery Task refactor – aims to prevent blocking the main thread while awaiting for the LAN discovery timeout; This is accomplished by moving the whole discovery functionality into its task and using a non-blocking timer to finish the task. Also fixes discovery sockets not being made non-blocking, which could cause the main thread to hang for very long periods of time every pre-frame.
- NETWORK/NETPLAY/TASKS: Find content task refactor – fixes many issues along the way, including a couple of nasty memory leaks that would leak thousands of bytes each time the task ran. It also expands the original concept by matching currently run content by filename (CRC matching is always performed first though).
- NETWORK/NETPLAY/TASKS: Find content task refactor – Ensure CRC32 is 8 characters long
- NETWORK/NETPLAY/LOBBY: Add setting for filtering out rooms with non-installed cores
- NETWORK/NETPLAY/LOBBY: Hide older (incompatible) rooms
- NETWORK/NETPLAY/LOBBY: Add a toggleable filter for passworded rooms. In addition, move lobby filters into its own submenu for better organization.
- NETWORK/NETPLAY/MENU: Chat supported info for the host kick submenu
- NETWORK/NETPLAY/MENU: Localize relay servers
- NETWORK/NETPLAY/MENU: Host Ban Submenu
- NETWORK/NETPLAY/MENU: Add client devices info to the kick sub-menu
- NETWORK/NETPLAY/MENU: Path: Netplay -> Host -> Kick Client – Allows the host to kick clients. Allows the host to view client information: connected clients (names), status (playing/spectating) and ping.
- NETWORK/NETPLAY/VITA: Add net_ifinfo support
- NETWORK/NETPLAY/VITA: Enable partial LAN discovery
- NETWORK/NETPLAY/VITA: Change default UDP port to 19492
- NETWORK/NETPLAY/VITA: Do not multiply negative timeout values
- NETWORK/NETPLAY/VITA: Fix epoll's timeout parameter
- NETWORK/NETPLAY/VITA: Launch compatibility patch
- NETWORK/NETPLAY/3DS: Launch compatibility patch
- NETWORK/NETPLAY/3DS: Adapt POLL for 3DS platform
- NETWORK/NETPLAY/PS3: Launch compatibility patch
- NETWORK/NETPLAY/WII: Enable net_ifinfo for some features. In practice, this only allows the netplay's UPnP task to succeed on the Wii.
- NETWORK/NETPLAY/WIIU: Launch compatibility patch
- NETWORK/NETPLAY/SWITCH: Launch compatibility patch
- NETWORK/UPNP: Attempt support for remaining platforms
- NETWORK/UPNP: Support for IPv4 tunneling
- ODROID GO2: Increase DEFAULT_MAX_PADS to 8 for ODROIDGO2, since that impacts the RG351[X] consoles. The RG351[X] have a USB host controller and can have an arbitrary number of USB gamepads.
- ONLINE UPDATER: Online Updater menu reorganizing
- OSX: Fixed items of system top menu bar on macOS
- OSX: Revision to macOS app icon set
- PLAYLISTS: Ensure history list will contain CRC32
- PLAYLISTS: Fix CRC32 comparison – as state->content_crc has "|crc" suffix.
- PS4/ORBIS: Orbis/PS4 Support using OrbisDev toolchain
- PS4/ORBIS: Update xxHash dependency
- PS4/ORBIS: Shader cache
- RETROFW: Exclude unused HAVE_HID for RetroFW
- RETROFW: Support battery indicator on RetroFW
- RETROFW: Enable menu toggle button on retrofw devices
- SHADERS: Shader Preset Loading of Multiple additional #references lines for settings
- SHADERS: Shader Load Extra Parameter Reference Files – this adds the ability to put additional #reference lines inside shader presets which will load additional settings. The first reference in the preset still needs to point at a chain of presets which ends with a shader chain, and subsequent #reference lines will load presets which only have parameter values adjustment. This allows presets to be made with a modular selection of settings. For example with the Mega Bezel one additional reference could point at a preset which contained settings for Night mode vs Day mode, and another reference could point to a preset which contained settings for how much the screen should be zoomed in.
- SHADERS/MENU: Increase shader scale max value
- SCANNER/DC: Fix Redump bin/cue scan for some DC games
- SCANNER/GC/WII: Add RVZ/WIA scan support for GC/Wii
- SCANNER/PS1: Improved success rate of Serial scanning on PS1 by adding support for the xx.xxx format
- SCANNER/PS1: Changed return value of detect_ps1_game function to actually return a failure when the Serial couldn't be extracted. Scanner will then fallback on crc check, and usually ends up finding the games in the database.
- SWITCH: Enable RWAV (WAV audio file) support
- STRING: Do not assume char is unsigned
- TASKS: More thread-awareness in task callbacks
- TASKS: Fix race condition at task_queue_wait
- TVOS: Revised tvOS icons w/ updated alien.
- VFS: Fix various VFS / file stream issues
- VULKAN: Fix more validation errors
- VULKAN: Attempt to fix validation errors with HDR swapchain. Always use final render pass type equal to swapchain format. Use more direct logic to expose if filter chain emits HDR10 color space or not
- VULKAN/ANDROID: Honor SUBOPTIMAL on non-Android since you'd want to recreate swapchains then. On Android it can be promoted to SUCCESS. SUBOPTIMAL_KHR can happen there when rotation (pre-rotate) is wrong.

- VULKAN/DEBUG: Automatically mark buffer/images/memory with names
- VULKAN/DEBUG: Move over to VK_EXT_debug_utils. Debug marker is deprecated years ago.
- VULKAN/HDR: Fix leak of HDR UBO buffer
- VULKAN/BFI: Fix BFI (Black Frame Insertion) regression
- WINDOWS: Fix exclusive fullscreen video refresh rate when vsync swap interval is not equal to one – refresh rate in exclusive fullscreen mode was being incorrectly multiplied by vsync swap interval, breaking swap interval functionality at the gfx driver level
- WIN32: Do optimization for Windows where we only update the title with SetWindowText when the previous title differs from the current title
- WIN32: Skip console attach when logging to file
- WIN32: Remove black margins with borderless non-fullscreen window
- WIN32/TASKBAR: Release ITaskbarList3 on failed HrInit – pointer wasn't NULL'd, thus set_window_progress would cause weird behavior
- WII/GX: Fix potential datarace
- WIIU: Implement sysconf and __clear_cache
- WIIU: Add OS memory mapping imports
- UWP: Added launch protocol arg 'forceExit' so a frontend can tell an already-running RetroArch UWP instance to quit.
- UWP: Enable core downloader/updater
- UWP: Remove copy permissions as its inefficient as we can just directly assign the new ACL and that works
- Xbox/UWP: Remove expandedResources
- Xbox/UWP: UWP OnSuspending crash fix
- Xbox/UWP: Enable savestate file compression by default for UWP/Xbox – got told there are no more issues with it
- Xbox/UWP: Add support for 4k to angle on xbox for MSVC2017 build

Blog (https://www.libretro.com/index.php/category/blog/), Libretro (https://www.libretro.com/index.php/category/libretro/), Netplay (https://www.libretro.com/index.php/category/netplay/), RetroArch (https://www.libretro.com/index.php/category/retroarch/), Vulkan (https://www.libretro.com/index.php/category/vulkan/)

RetroArch – Introducing the Mega Bezel Reflection Shader (https://www.libretro.com/index.php/retroarch-introducing-the-mega-bezel/)

July 8, 2022 (https://www.libretro.com/index.php/retroarch-introducing-the-mega-bezel/)

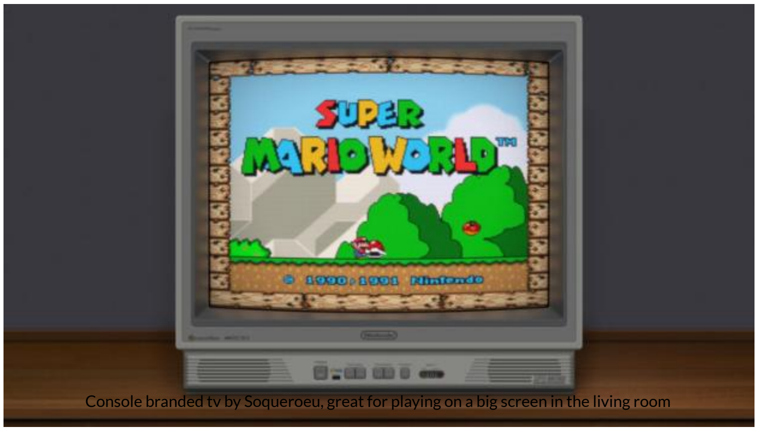
Article written by HyperSpaceMadness

RetroArch keeps introducing innovations to the retrogaming world, constantly building simple roads for players to enjoy classic games in new and sometimes better ways.

Getting the sweet spot between ease of use and customization can be a time-consuming process, and sometimes requires a deep insight of how old technologies worked: refresh rate, aspect ratio, scaling, overscan, deconvergence are terms which we may or may not be familiar with, and these all play an important part in building a retro gaming experience that feels better, yet passionately authentic.

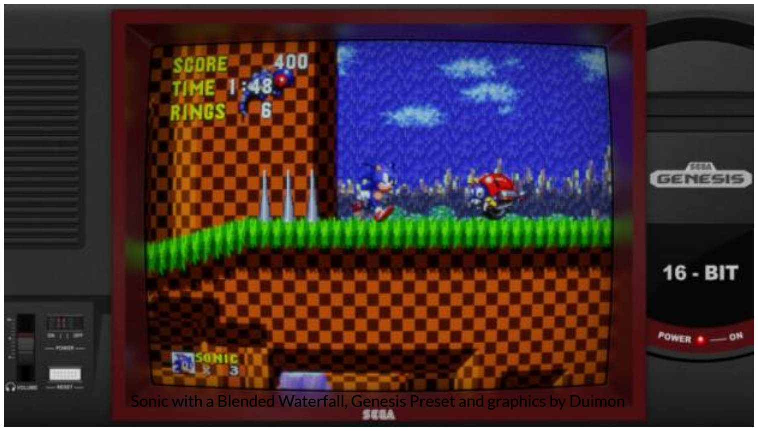
How do we get a handle on this?

Building an idealized CRT (cathode ray tube) like display experience. Getting the best out of post-processing with the latest CRT shaders fused into a "one stop" solution. Making it incredibly easy to customize, and yet performant. A fresh and unique starting point for the retro game lover.



Console branded tv by Soqueroeu, great for playing on a big screen in the living room

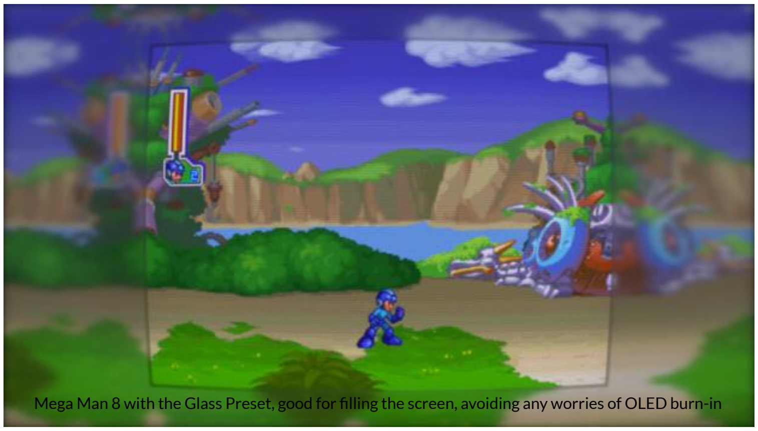
The Mega Bezel Project started back in July 2019 when developer HyperspaceMadness was looking at experimental shaders creating real-time reflections on emulated display bezels. More than two years later, the swiss-army-knife of visual simulation to enhance the retro game experience is ready for players!



Sonic with a Blended Waterfall, Genesis Preset and graphics by Duimon

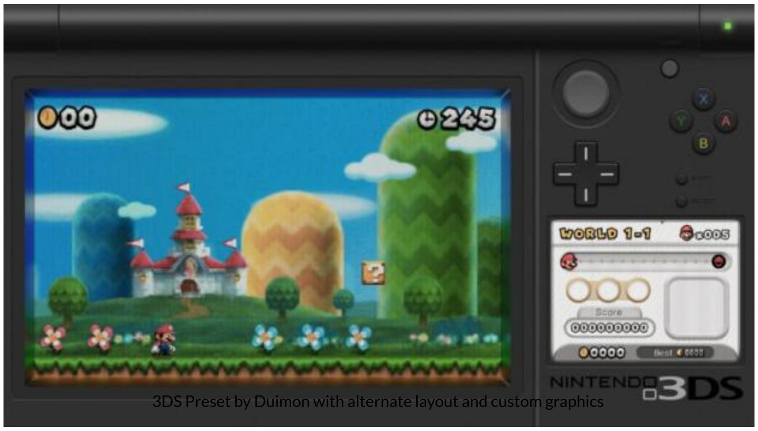
The Mega Bezel is unique in that it bends the common definitions of shaders and overlays in an out-of-the-box experience: custom calculations take care of games native resolution and scaling, dynamically draw bezels around the gameplay area filled with curvature simulation and reflections, incorporating a unique pipeline of CRT simulation models and other visual conditioning of the game image, color correction, de-dithering, and adding responsive backgrounds and lots of additional features to enjoy.

The shader centralizes a lot of complex tasks and makes them instantly available for all cores: screen rotation and position, horizontal and vertical orientation, zooming, cutting away games black spaces to get a real full-screen, and filling the aspect ratio difference between the emulated screen and your monitor with interesting graphics. Mega Bezel even adds original solutions, like the 'Night Mode' to simulate a dimly lit room and 'Glass Mode' for a modern and dynamic way to fill the entire monitor.



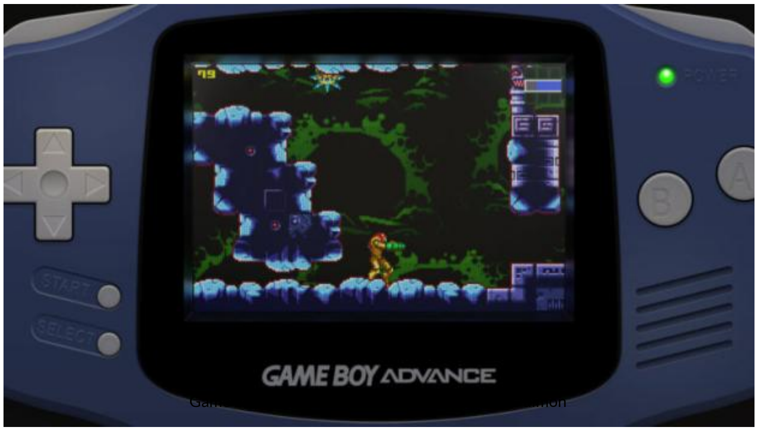
Mega Man 8 with the Glass Preset, good for filling the screen, avoiding any worries of OLED burn-in

Being based on contributions and discussions from the Libretro forums, Mega Bezel is a community project at its heart: shader writers and artists are actively developing features and customized presets which max out the shader capabilities, and making them freely available for retro players to enjoy and further customize, chasing the their ideal setup.



3DS Preset by Duimon with alternate layout and custom graphics

Easy to use for newcomers, deep in customization for emulation maniacs, flexible for artists: the Mega Bezel project is a fun ongoing journey that strives to bring wonderful features to everyone, minus the hassle of setup!



Gam[illegible]tion

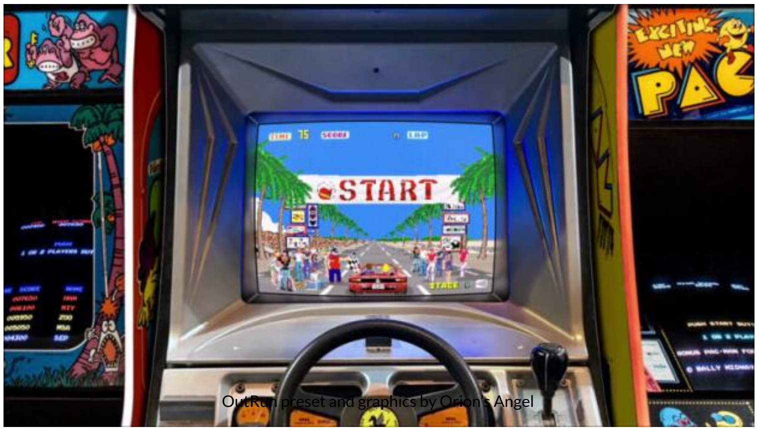
You can get the basic Mega Bezel shaders inside RetroArch by running **Online Updater -> Update Slang Shaders** then the shaders will be located inside **shaders/shaders_slang/bezel/Mega_Bezel/Presets**. Be sure to read the setup portion of the ReadMe.md to help you get started you can find it in the Mega_Bezel folder just mentioned.

A small group of talented artists has also come together using the Mega Bezel to create suites of shader presets with beautiful graphics covering many consoles, computers and PVMs to share with retro gamers.

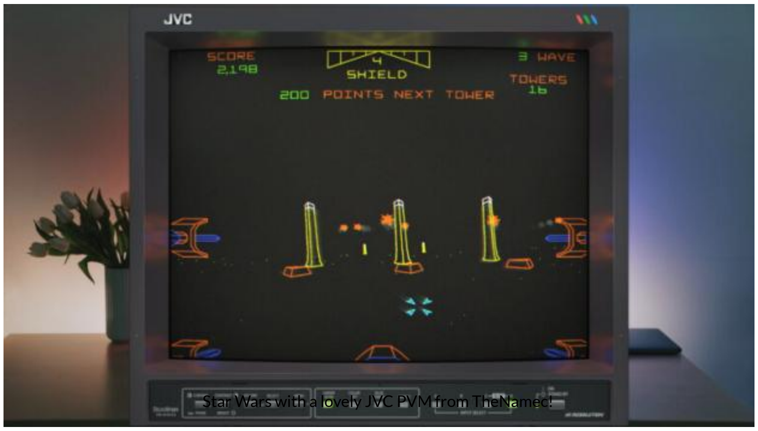
If this sounds exciting head over to the Mega Bezel thread on the Libretro forum to find more about the Mega Bezel, and links to the artist's pages and shader preset collections.

https://forums.libretro.com/t/hsm-mega-bezel-reflection-shader-feedback-and-updates/25512 (https://forums.libretro.com/t/hsm-mega-bezel-reflection-shader-feedback-and-updates/25512)

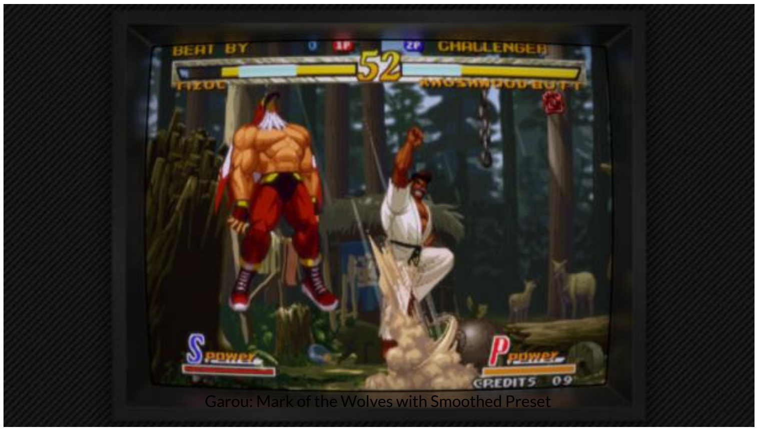
Below are more examples of the Mega Bezel and artwork from the community.



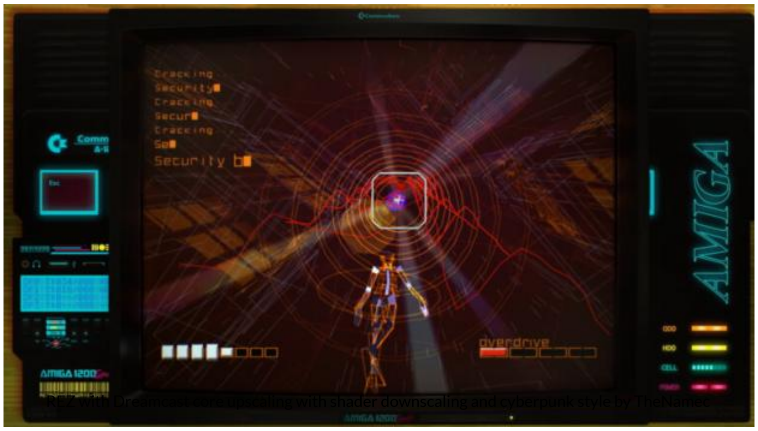
OutRun preset and graphics by Orion's Angel



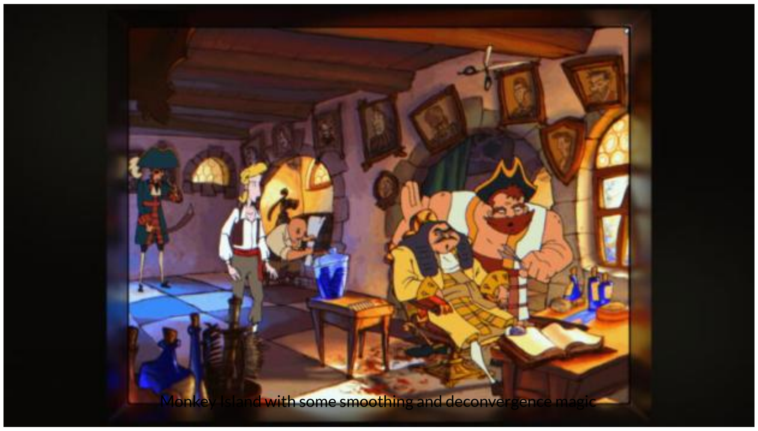
Star Wars with a lovely JVC PVM from TheNamec!



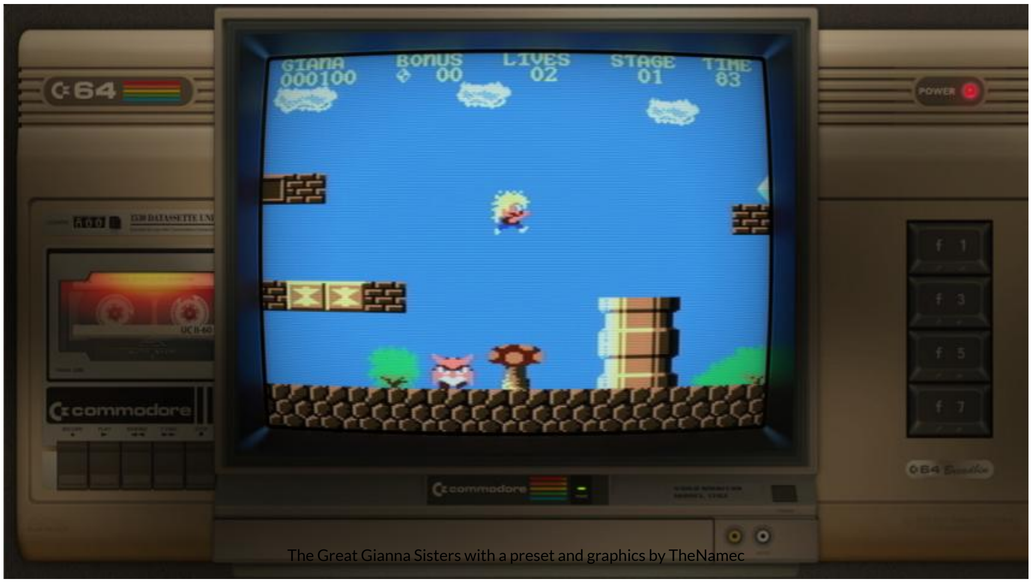
Garou: Mark of the Wolves with Smoothed Preset



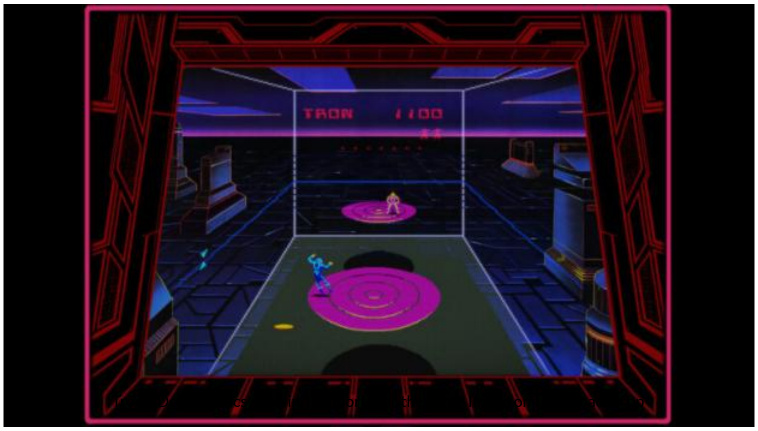
REZ with Dreamcast core upscaling with shader downscaling and cyberpunk style by TheNamec

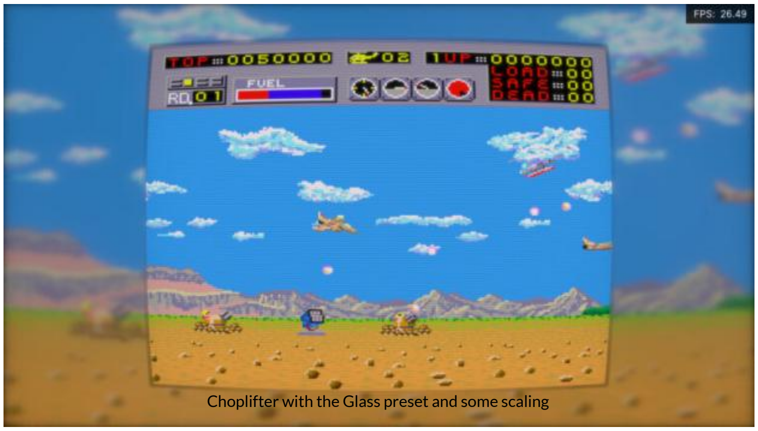


Monkey Island with some smoothing and deconvergence magic

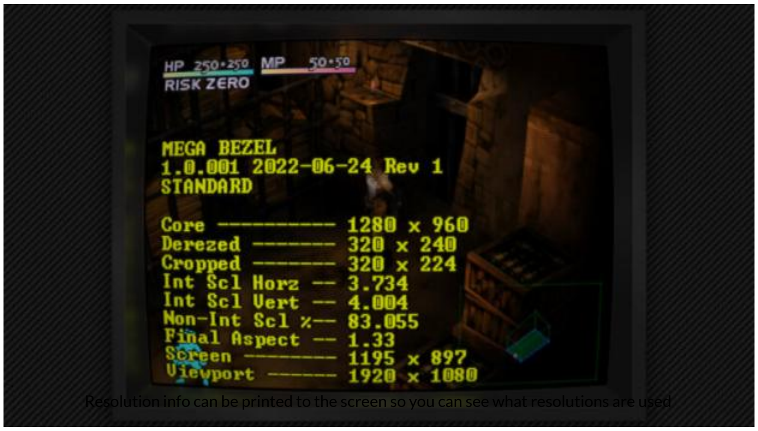


The Great Gianna Sisters with a preset and graphics by TheNamec





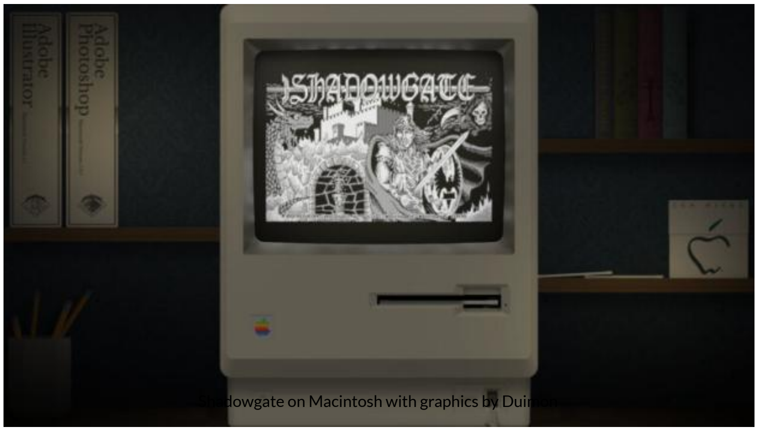
Choplifter with the Glass preset and some scaling



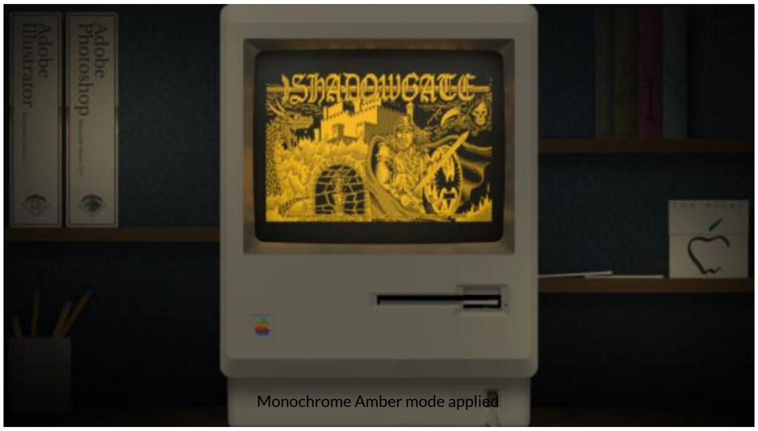
```
MEGA BEZEL
1.0.001 2022-06-24 Rev 1
STANDARD

Core ----------- 1280 x 960
Derezed -------- 320 x 240
Cropped -------- 320 x 224
Int Scl Horz --- 3.734
Int Scl Vert --- 4.004
Non-Int Scl %--- 83.055
Final Aspect --- 1.33
Screen --------- 1195 x 897
Viewport ------- 1920 x 1080
```

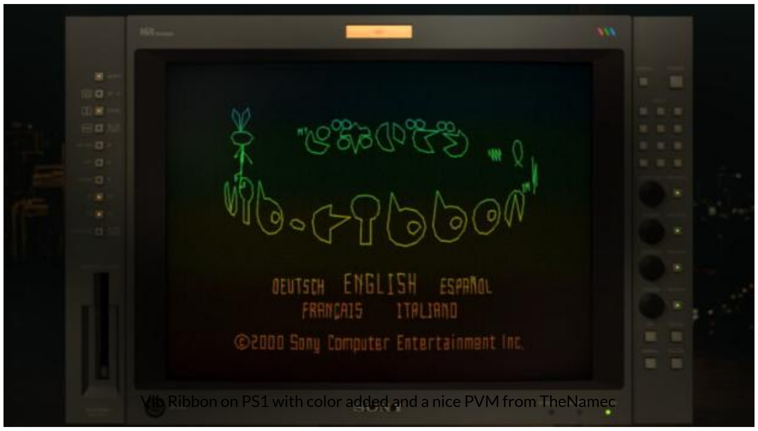
Resolution info can be printed to the screen so you can see what resolutions are used



Shadowgate on Macintosh with graphics by Duimon



Monochrome Amber mode applied



Vib Ribbon on PS1 with color added and a nice PVM from TheNamec

Blog (https://www.libretro.com/index.php/category/blog/), Mega Bezel (https://www.libretro.com/index.php/category/mega-bezel/), PlayStation1 (https://www.libretro.com/index.php/category/playstation1/), RetroArch (https://www.libretro.com/index.php/category/retroarch/), Shaders (https://www.libretro.com/index.php/category/shaders/), Slang (https://www.libretro.com/index.php/category/slang/)

RetroArch now on Windows Package Manager! (https://www.libretro.com/index.php/retroarch-now-on-windows-package-manager/)

June 27, 2022 (https://www.libretro.com/index.php/retroarch-now-on-windows-package-manager/)

Windows 10/11 users now have an even more convenient way of installing RetroArch! RetroArch is now available from the Windows Package Manager (see here (https://winstall.app/apps/Libretro.RetroArch)).

- Stables are identical to the ones found on our website/buildbot.
- They are digitally signed, meaning they pass the Windows Defender SmartScreen Protection. You will not get a warning from SmartScreen when attempting to install RetroArch this way. Useful if you are in a protected environment where your system administrator has locked down your ability to install unsigned third-party applications on your system.

The video will show you how to install it.

First, start up the Command Prompt. You can go the Start Menu and type in ‘Command Prompt’ and then click on it to start it up, or simultaneously press Windows key + R key, and then type in ‘cmd’.

How to search for RetroArch

Once on the command line, you can search for packages to see if they exist. To search for RetroArch, tyep in the following:

```
winget search RetroArch
```

Lowercase 'retroarch' will work as well. An entry should show up.

How to install RetroArch

Now that we know the package exists on the package manager, we should be able to install this. Simply type on the commandline:

```
winget install RetroArch
```

It will now install RetroArch without requiring any user interaction. Note that this version of RetroArch will not attempt to install the DirectX9 SDK in case it doesn't already exist. We assume on modern Windows you will use the Direct3D 10/11/12 or Vulkan/OpenGL drivers anyway.

Once installed, you should be able to find it from the Start Menu as a recently added application.

How to uninstall RetroArch

Uninstalling RetroArch once installed is similarly easy. Simply type the following on the commandline:

```
winget uninstall RetroArch
```

RetroArch's presence on app stores

Our aim with RetroArch is to be available on as many storefronts and outlets as possible. We have made some impressive progress over the years.

- RetroArch is available on Steam for Windows and SteamOS/Linux (see here (https://store.steampowered.com/app/1118310/RetroArch/)).
- RetroArch is available on the Windows Package Manager for Windows 10/11 (see here (https://winstall.app/apps/Libretro.RetroArch)).
- RetroArch is available on the Google Play Store for regular Android devices (see here (https://play.google.com/store/apps/details?id=com.retroarch)). A Plus version with additional features is also available here (https://play.google.com/store/apps/details?id=com.retroarch.aarch64).
- RetroArch is available on the Huawei AppGallery for Huawei branded Android devices (see here (https://appgallery.huawei.com/#/app/C104593727)).
- RetroArch is available on the Samsung Galaxy Store for Samsung branded Android devices (see here (https://galaxystore.samsung.com/detail/com.retroarch?hit_type=Desktop&source=GBadge_01_4415800_retroarch)).
- RetroArch is available on the Amazon App Store for Amazon branded Android devices (see here (https://www.amazon.com/dp/B09753XRVF)).
- RetroArch is available on itch.io (see here (https://retroarchofficial.itch.io/retroarch)).
- RetroArch is available on Canonical Snapcraft (see here (https://snapcraft.io/retroarch)).
- RetroArch is available on Flathub (see here (https://flathub.org/repo/appstream/org.libretro.RetroArch.flatpakref)).
- RetroArch has a Humble Bundle page (see here (https://www.humblebundle.com/g/retroarch)).

We have even more plans to expand that we cannot talk about yet!

Blog (https://www.libretro.com/index.php/category/blog/), Windows (https://www.libretro.com/index.php/category/windows/), Windows 10 (https://www.libretro.com/index.php/category/windows-10/), Windows 11 (https://www.libretro.com/index.php/category/windows-11/), Windows Package Manager (https://www.libretro.com/index.php/category/windows-package-manager/)

RetroArch now on the Samsung Galaxy Store! (https://www.libretro.com/index.php/retroarch-now-on-the-samsung-galaxy-store/)

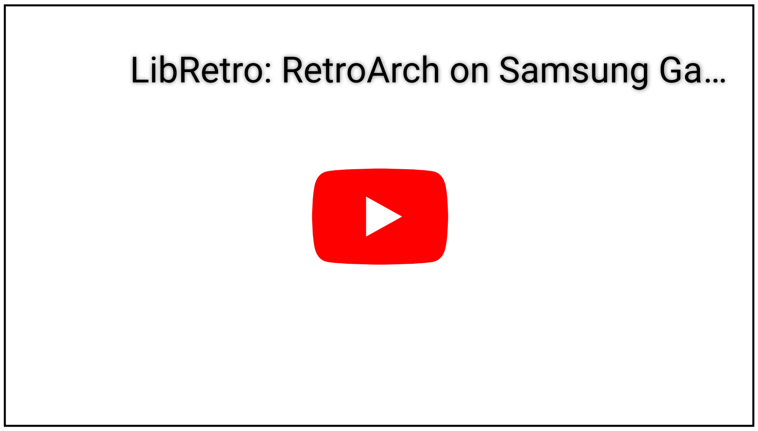
May 31, 2022 (https://www.libretro.com/index.php/retroarch-now-on-the-samsung-galaxy-store/)

Available on Galaxy Store

(https://galaxy.store/retroarch)
We keep expanding!

We're happy to announce that RetroArch is now available on the Samsung Galaxy Store! For free, of course.

Q&A



How can I download RetroArch from the Galaxy Store?

The Galaxy Store should be pre-installed by default on Samsung phones. It is a storefront available exclusively for Samsung-branded devices.

Open the app on your device, type in RetroArch in the search bar, and download it from there.

How does this version differ from the Google Play Store version?

The version available on the Galaxy Store is identical to the version you can download from our website.

It has a couple of big advantages over the Google Play Store version:
* There is no set core limit. For RetroArch Plus on the Play Store, it is up to 127 cores that can be installed. We have to hand-pick these cores specifically so that users can install them on the Google Play Store.
* Cores are distributed from the Libretro buildbot infrastructure. There are far more cores available than on the Play Store.

Should I use this version or the Google Play Store version?

On a Samsung phone, you have the choice to choose between either version. Regardless, we highly recommend you use the Galaxy Store version over the Google Play Store version. Reasons are listed above, but on top of that, the Google Play Store version has not been updated for a while because of increasing restrictions and requirements that we haven't caught up with yet.

Bottom line, we anticipate the Google Play Store version to become more and more nerfed as time goes on unfortunately. There is nothing we can do about this, these are restrictions and limitations imposed by Google to have the software available for distribution on the Play Store. To get a more full-featured version, download the Galaxy Store version.

RetroArch should now be available on the Google Play Store (https://play.google.com/store/apps/details?id=com.retroarch), Amazon App Store (https://www.amazon.com/dp/B09753XRVF), Huawei App Gallery (https://appgallery.huawei.com/#/app/C104593727), and Samsung Galaxy Store (https://galaxy.store/retroarch). No matter what device you are on and which ecosystem you are in, we try to have you covered.

Android (https://www.libretro.com/index.php/category/android/), Blog (https://www.libretro.com/index.php/category/blog/), Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/), Samsung Galaxy (https://www.libretro.com/index.php/category/samsung-galaxy/), Samsung Galaxy Store (https://www.libretro.com/index.php/category/samsung-galaxy-store/)

Lakka 4.2 release (https://www.libretro.com/index.php/lakka-4-2-release/)

April 28, 2022 (https://www.libretro.com/index.php/lakka-4-2-release/)

New version of Lakka has been released!

We are happy to announce the new and updated version of Lakka. Read the full article here (https://lakka.tv/articles/2022/04/27/lakka-4.2/).

Lakka (https://www.libretro.com/index.php/category/lakka/), Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

RetroArch 1.10.3 release! (https://www.libretro.com/index.php/retroarch-1-10-3-release/)

April 15, 2022 (https://www.libretro.com/index.php/retroarch-1-10-3-release/)

RetroArch 1.10.3 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

Core Updates

FBNeo, LRMAME2003 Plus, UAE and VICE have seen the regular weekly updates/improvements. We can't list all the changes there, so we just suggest you go to the respective Github repositories and check out the chanes there.

LRMAME

LRMAME updated to version 0.242 (latest). LRMAME is now also available for ARM Macs now. You can get it from the Core Downloader.

Nestopia

FourScore support (4 player multitap) has been added for the following games:

- Spacey McRacey
- NNNNNN
- Arkade Rush
- Justice Duel
- BMX Simulator
- Way of the Exploding Fist

PCSX ReARMed

This core has a new auto frameskip mode (based on free audio buffer space).

The Lightrec dynamic recompiler has been updated, and it should fix several crashes and bugs that occurred before. This would only affect users on x86/x86_64 and MIPS architecture processors, as ARM architecture-based systems continue to use the Ari64 dynarec instead.

There has been a GunCon overhaul, the following input descriptors have been added:

- Add Trigger, Reload, Aux A, and Aux B as mappable buttons in RetroArch menu for players 1 and 2.
- GunCon trigger, A, and B buttons are mapped to Gun Trigger, Gun Aux A, and Gun Aux B instead of hard coded to left click, right click, and middle click.
- Force cursor to corner of screen for offscreen reload so that reloading works on all four edges of the screen.
- Allow Gun Reload RetroArch input to emulate an offscreen shot.
- Switch gun coordinates from "Pointer" type to "Lightgun" type.

GW (Game & Watch)

The **GW (Game & Watch) Libretro core** is now available for the MSVC 2005 and 2010 Windows versions. These versions can run on older Windows OS versions than the regular version.

gpSP

The **gpSP Libretro core** now uses a small translation cache for the Miyoo platform.

Cap32

An emulator of the Amstrad CPC 8bit home computer range. This has seen several improvements.

- DB: new games from retroachievements
- UI: added DB icon DSK to status bar
- DB: you could add direct tokens using $ (for joystick keybinds or cleans)
- DB: added DB v1 using clean-cpc-db info
- CORE: added model 664 to allow DSK and BASIC 1.0
- CORE: detect some configurations from filename
- VIDEO: minor fixes (requires more work)

SMS Plus GX

The **SMS Plus GX Libretro core** should now be more stable on RetroArch PSP. We achieve this by avoiding unaligned memory access. Previously, after starting a game, the console would have a tendency to locks itself and shut down.

Other related changes – we replaced ALIGN_LONG with ALIGN_DWORD for Miyoo and RetroFW to match
the standalone versions. This fixes Master System background rendering. It was dropped from 3DS as ARMv6 allows unaligned memory access and defining that macro had no effect anyway. ALIGN_DWORD was dropped from Raspberry Pi (ARMv6/7/8), Classic (ARMv7), OS X non-PPC (x86, ARMv8), Vita (ARMv7) and Switch (ARMv8) as those platforms support unaligned memory access.

Beetle Virtual Boy

Fixed a couple inaccuracies in the VSU modulation emulation, fixing a few sound effects in "Virtual Boy Wario Land".

Mesen

The Nintendo Entertainment System emulator core has seen a couple of improvements.

Before, the core would upload audio by using the audio batch callback multiple times per frame, unduly 'stressing' the frontend audio buffer and leading to poor AV synchronisation.

We now ensure that the audio batch callback is only used once per frame (unless the frontend does not support batches of sufficient size, in which case the samples will be split appropriately).

We also did the following:

- Sets the default audio sample rate to 48000 Hz. The previous default of 96000 Hz is so high that RetroArch is required to flush the audio driver twice per frame, which is bad for AV synchronisation.
- Removes the 192000 and 384000 sample rate options, since these are in fact unsupported by the underlying emulator code…

Add 4:3 (Preserved) & 16:9 (Preserved) aspect ratios
Mesen by default preserves the aspect ratio in all cases when cropping the overscan, which results in a difference between the core provided 4:3 and 16:9 ARs, and RetroArch's own 4:3 and 16:9 ARs, which doesn't always results in a ideal image (specifically 16:9 on a 16:9 display will look weird when cropping is applied).

We now separate Mesen's preserved 4:3 and 16:9 ARs into their own selections for the core provided aspect ratio so people can choose whenever or not they want the aspect ratio to be preserved when using either one of the selections as their core provided aspect ratio.

bsnes Mercury/bsnes C++98

The Super Nintendo Entertainment System emulator core has seen a couple of improvements.

Before, the core(s) would upload audio in packets of 64 samples – which means the audio batch callback is used multiple times per frame, unduly 'stressing' the frontend audio buffer and leading to poor AV synchronisation.

We now ensure that the audio batch callback is only used once per frame.

REminiscence

This **Flashback game engine core** has now been added for the Miyoo platform as well.

ScummVM

Several serious crashes should be fixed now as a result of us updating the libco coroutines middleware library.

FCEUmm

This **Nintendo Entertainment System** emulator core has seen several improvements.

More mapper additions and improvements
Improve mappers 49, 215/258, 340, 341, 351 and 444. Add newly-(re)assigned mappers 294 and 310. Add new mapper 467.

Expose internal audio RF filter option
The core already contains a low pass audio filter designed to recreate the 'muted' sound of the NES when connected to a television via the RF modulator – but for some reason this functionality is not enabled/exposed.

We have simply wired it up to a new Audio RF Filter core option. When enabled, the (subjective) improvement in audio quality is quite dramatic. The filter has a negligible performance impact.

(This filter produces the effect discussed here: https://forums.libretro.com/t/lowpass-filtering-for-nes-rf/37258 (https://forums.libretro.com/t/lowpass-filtering-for-nes-rf/37258))

Add optional 'fake' stereo sound effect
We added a new Stereo Sound Effect core option which may be used to simulate stereo sound by delaying the right audio channel (relative to the left) when upmixing the mono output from the NES. The delay can be configured from 1 to 32 ms.

The effect is identical to the fake stereo currently available in the Mesen core.

minivmac

minivmac is an emulator for the Mini vMac, a miniature Macintosh. We added this core now for ARM Macs. It can be downloaded from the Core Downloader.

Genesis Plus GX

Genesis Plus GX is a **Sega Master System/Sega Game Gear/Sega Megadrive/Sega Genesis emulator core**.

We are using the low memory codepath now for Miyoo systems. As this platform only has 32MB RAM, like the RS-90.

xRick

The **Rick Dangerous game engine core** has been added for the Miyoo platform.

Snes9x 2005

This **Super NIntendo Entertainment System emulator core** has seen several improvements.

Before, the core had bad audio sample pacing:

- Neither variant of the core sent a number of samples per frame that would match the nominal expected values given by the sample rate and fps set in retro_get_system_av_info()
- Due to integer rounding errors, the non-plus core always would send too few samples
- The 'plus' version of the core would send the 'correct' number of samples, in terms of actual emulation – but this does not tally with the sample rate reported to the frontend. Moreover, the 'plus' core would call the audio batch callback twice per frame, which unduly stresses the frontend audio buffer.

As a result, the core had bad audio/video synchronisation, affecting frame pacing.

We fixed several issues:

- The audio sample rate is now reported as 32040 Hz
- The non-plus core uses an accumulator to ensure that 'fractional' audio samples are accounted for and sent when required
- The plus core now uploads audio samples only once per frame

In addition, we did the following

- Fixed three memory leaks that were found in the core
- Modified the Console Region core option to require a restart (since it has never been possible to change this at runtime…)

Snes9x2005 Non-Plus: Add optional low pass audio filter
Apart from a substantial difference in audio emulation accuracy, probably the most obvious difference between the 'plus' and 'non-plus' versions of the core is that the latter has an inadequate level of low pass audio filtering, leading to tinny/scratchy sound.

We added a simple optional low pass filter at the output stage of the 'non-plus' core. When enabled, audio is more mellow/bassy, and the generated sound is closer to that produced by the 'plus' version – with only a negligible increase in performance requirements.

Snes9x 2010

This **Super NIntendo Entertainment System emulator core** has seen several improvements.

Use audio batch callback only once per frame

Before, the core would upload samples in batches of ~64, which means the audio batch callback is used many (~9) times per frame. This 'overstresses' the frontend audio buffer and leads to bad AV synchronisation.

We have fixed the issue by ensuring that the audio batch callback is used to send all available samples only once per frame.

Improve save state efficiency + fix save state size

At present, every time that retro_serialize_size() is called (i.e whenever save states are used), the core determines the save state size by allocating a temporary 5 MB buffer and writing into this an actual save state. Moreover, it then fails to report the actual size correctly due to a bug in the memory stream wrapper code – which means save states are always 5 MB in size. This represents a terrible inefficiency.

Now, the save state size is now calculated independently of regular save state creation. No temporary buffer is required, and there is no need to actually write a save state to memory – and save states now have the correct size (~830 kb)

SwanStation

This **Sony PlayStation1 emulator core** has been updated.

- Remove 'Force Pop'n Mode' & 'NeGcon Steering Axis Deadzone' options

60Hz modes for > 60Hz emulated platforms

Big improvements for WonderSwan, Lynx and PokeMini emulator cores for the majority of systems that don't happen to have VRR displays!

Beetle WonderSwan

At present the core runs at ~75Hz, matching the native refresh rate of the WonderSwan hardware. This is fine if the core is run on a VRR display (or one that natively supports 75Hz…), but on regular 60Hz panels it can cause issues. In particular, screen tearing is very likely to occur. You can experience this on Linux (when not using a compositor and without vsync forced at the driver level) and on 3DS. The tearing is so bad on 3DS that we would previously consider the core to be unusable on that platform…

We now added a new 60Hz Mode core option, which can be used to force the core to run at 60Hz (actually 60.38Hz, but RetroArch handles this nicely via dynamic rate control). Note that the core still runs at the 'correct' speed when this option is enabled – internally, the core is running the nominal ~75 frames per second, but every 5th frame is 'dropped'. This reduces video smoothness, but then 75Hz on a 60Hz display is not smooth either. More importantly, enabling this option eliminates screen tearing.

In addition, we have also made the following minor changes:

- The frontend reported framerate is now set correctly in 75Hz mode (previously this was truncated, leading to a slight tendency for the frontend audio buffer to under-run)

- The internal audio samples buffer has been reduced from a ~64kb (!) static array to a tiny, dynamically created array of just the correct size
- On 3DS, the video buffers are now allocated in linear memory (for improved performance)
- The 96000, 192000 and 384000 audio sample rate options have been removed, because they are nonsensical and harm AV synchronisation

Thanks to this 60Hz mode, Beetle WonderSwan is now perfectly playable on RetroArch 3DS. We have enabled this option by default. If you are using a VRR display or if you are running at a native 75Hz resolution and would like to change it back to the native refresh rate, you can just turn this option off in Quick Menu -> Options.

We have also added the core for RetroArch PS2, although it can't reach fullspeed. It's debatable whether it's worth including, but for now we keep it in.

There is also a new optional audio feature. The WonderSwan has a tendency to produce rather harsh/abrasive chiptunes. The low pass audio filter softens and 'mellows out' the generated sound.

PokeMini

At present the core runs at 72Hz, matching the native refresh rate of the Pokemon Mini hardware. This is fine if the core is run on a VRR display (or one that natively supports 72Hz…), but on regular 60Hz panels it can cause issues. In particular, screen tearing is very likely to occur. We could experience this on Linux (when not using a compositor and without vsync forced at the driver level) and on 3DS.

We have now added a new 60Hz Mode core option (enabled by default), which can be used to force the core to run at 60Hz. Note that the core still runs at the 'correct' speed when this option is enabled – internally, the core is running the nominal 72 frames per second, but every 6th frame is 'dropped'. This reduces video smoothness, but then 72Hz on a 60Hz display is not smooth either (and few Pokemon Mini games are 'smooth' to begin with…). More importantly, enabling this option eliminates screen tearing.

Handy

This Atari Lynx emulator core has seen several big improvements.

Fix frame pacing
Before, this core had entirely broken frame pacing. The core reported a fixed refresh rate of 75Hz to the frontend, but the Lynx (and the internal emulation code) has a variable refresh rate of 0-75Hz; games can render at any rate they please. In retro_run(), the Lynx is always emulated until the next 'end of frame' event occurs – if a game renders at e.g. 25 fps, this means retro_run() will actually correspond to (1/25) seconds worth of Lynx runtime instead of the expected (1/75) seconds. In this case, the game is emulated too quickly – but it appears to run at the correct speed in the frontend because the core uploads an 'oversized' audio buffer (1/25 seconds worth of samples). RetroArch syncs on audio in such a way that when too many samples are received, the frontend runs in 'slow motion' – so the 'too fast emulation' + 'too many audio samples' effectively cancel out. But the results are awful. This is a significant violation of the libretro API, and it destroys the frontend's ability to properly synchronise audio and video, and to pace the frames correctly.

We now modified the run loop such that a fixed number of CPU cycles are emulated on each call of retro_run(), corresponding to the actual frontend output video refresh rate (which can be set via a new Video Refresh Rate core option). Thus the Lynx is always emulated at the correct speed, audio is always uploaded in batches of the correct size, and generated video frames are captured and output when available (and when the frontend can accept them).

The default Video Refresh Rate has been set to 60Hz, which provides smooth results for most games (and also eliminates screen tearing on 60Hz displays, which was an issue when the core only reported a 75Hz refresh rate). If a game has a higher frame rate than this (rare, but e.g. the intro and menus of California Games run at the full 75 fps), then 'excess' frames will be dropped. Users with 75Hz+ VRR displays can set higher refresh rates to improve video smoothness in these cases.

Improve save state efficiency
Before, the retro_serialize() function determines the save state size by allocating a temporary ~310kb buffer, writing an actual save state into it, then fetching the resultant buffer occupancy. This is terribly inefficient – and retro_serialize() is called 3 times every time a state is saved or loaded…

We modified the serialisation memory stream code to allow a 'virtual' save state to be made – no buffer is required, and no data are copied. This means retro_serialize() can now fetch the save state size with no memory allocations and no wasted effort.

Add optional LCD ghosting filter

We added a new LCD Ghosting Filter core option which can be used to apply an LCD ghosting effect by blending multiple successive frames. The number of blended frames can be set from 2-4; using more frames improves the quality of the effect at the expense of increased performance requirements.

LCD ghosting is particularly beneficial for the Lynx because many games run at very low frame rates, and some blurring helps to smooth out the frequently 'jerky' screen updates.

RetroArch Updates

See the Changelog below for a detailed breakdown of all the changes that have happened.

One of the biggest changes for Steam users by far is the new Steam Discord Rich Presence support. NOTE: You will need to use the desktop client in order for this to work. It won't work with the webbrowser client.

Changelog

1.10.3

- ANDROID: Decouple Play Core dependency to bring app into compliance for F-Droid
- AI/SERVICE: Disable AI Service setting by default
- BLUETOOTH/LAKKA: bluetoothctl: add / modify pairing steps
- CHEEVOS: Disallow manual frame delay setting in Hardcore Mode
- DATABASE: Serial scanning for Wii now includes WBFS
- INPUT/MAPPING: Fix offset + crash when clearing input port binds
- INPUT/MAPPING: Fix saving of 'Analog to Digital Type' when configuration overrides are used
- LOCALIZATION: Add Valencian language option
- LOCALIZATION: Updates
- MENU/SETTINGS: Move 'Show Menu Bar' under 'Windowed Mode' settings
- MENU/SETTINGS: Add sublabels for 'Subsystems' and 'Input Deadzone/Sensitivity'

- MENU/SETTINGS: Move 'On-Screen Notifications' to top
- MENU/XMB: Unified the shadow alpha value to a slightly darker one for better readability
- MENU/XMB: Corrected the option label and sublabel for actual behavior
- MIYOO: Enable ALSA audio driver and default to it
- PSP: Take out extra languages/localization, adds about 4/5MB to the binary, and RAM is limited on PSP (32MB and 64MB RAM models)
- STATIC PLATFORMS: Populate all history list metadata when launching content from playlists
- STEAM: Introduce Steam Rich Presence
- VIDEO: Fast-Forward Frameskip improvement
- VIDEO/THREADED: Stability fixes
- WINDOWS/WINRAW: Fix multiple light guns
- WIIU: Fix USB get_device_name(), don't truncate to three chars

Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

Lakka 4.1 release (https://www.libretro.com/index.php/lakka-4-1-release-2/)

April 8, 2022 (https://www.libretro.com/index.php/lakka-4-1-release-2/)

New version of Lakka has been released!

We are happy to announce the new and updated version of Lakka. Read the full article here (https://lakka.tv/articles/2022/04/08/lakka-4.1/).

Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

Lakka 4.1 RC release (https://www.libretro.com/index.php/lakka-4-1-release/)

April 3, 2022 (https://www.libretro.com/index.php/lakka-4-1-release/)

New version of Lakka has been released!

We are happy to announce the new and updated version of Lakka. Read the full article here (https://lakka.tv/articles/2022/04/03/lakka-4.1-rc/).

Blog (https://www.libretro.com/index.php/category/blog/), Lakka (https://www.libretro.com/index.php/category/lakka/), Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

RetroArch 1.10.2 release! (https://www.libretro.com/index.php/retroarch-1-10-2-release/)

March 29, 2022 (https://www.libretro.com/index.php/retroarch-1-10-2-release/)

RetroArch 1.10.2 has just been released.

Grab it here (http://retroarch.com/?page=platforms).

If you'd like to learn more about upcoming releases, please consult our roadmap here (https://github.com/orgs/libretro/projects/1).

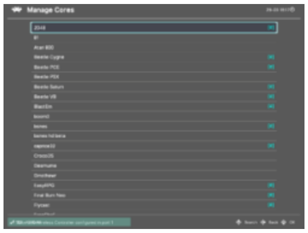
Remember that this project exists for the benefit of our users, and that we wouldn't keep doing this were it not for spreading the love to our users. This project exists because of your support and belief in us to keep going doing great things. We have always prioritized the endusers experience, and unlike others, we have never emburdened them with in-app ads, monetization SDKs or paywalled features, and we intend to continue to do so. If you'd like to show your support, consider donating to us. Check here (http://retroarch.com/index.php?page=donate) in order to learn more. In addition to being able to support us on Patreon (https://www.patreon.com/libretro), there is now also the option to sponsor us on Github Sponsors (https://github.com/sponsors/libretro)! You can also help us out by buying some of our merch on our Teespring store (https://teespring.com/stores/retroarch)!

Highlights

Big improvements for Steam and Steam Deck users

We're gearing up for Steam Deck, and in the process we are finally starting to turn the RetroArch Steam version into something more than just a plain port (courtesy of Mats).

We came up with a SteamWorks shim that allows RetroArch Steam to interface with the Steamworks API. Mist, our middleware tool, runs in a separate process, runs concurrently wtih RetroArch Steam, and functions as a bridge between this separate process interfacing with Steamworks and the GPL application itself running in an entirely different process. This is 100% GPL compliant and the same approach has been employed by numerous other examples on Steam, including the aforementioned Icculus.



Manage Cores

2048
81
Atari 800
Beetle Cygne
Beetle PCE
Beetle PSX
Beetle Saturn
Beetle VB
BlastEm
boom3
bsnes
bsnes hd beta
Caprice32
CrocoDS
DeSmuME
DinoThawr
EasyRPG
Final Burn Neo
Flycast

Thanks to Mist, here are some of the big new features for Steam users (and in particular, Steam Deck users):

- You can now install and uninstall cores directly from inside RetroArch by going to 'Manage Cores' from inside the menu. No longer do you have to manually browse Steam with your webbrowser and download random core DLCs for RetroArch, you can do it from within the convenience of the app running itself.
- On Steam Deck, the native OSK (Onscreen Keyboard) will appear now instead of the RetroArch default OSK. This should give a more seamless experience. We have decided against enabling the Steam OSK with Big Picture since it seems partially broken and instantly dismisses itself (or at least those were our testing results on a Steam Deck).

As a result of these changes, the Steam port is starting to become its own thing rather than just a straightforward no-frills port of the Windows/Linux versions.

Improved audio sample pacing / latency in many cores

Big improvements have been made to several cores concerning improved audio latency and audio sample pacing.

Not only should performance be better, but you should also be able to lower audio latency buffers now while still getting perfect sound.

Here are some of the cores that have received work recently on this front:

- Cannonball
- Flycast
- Gambatte
- Nestopia
- Snes9x
- Snes9x 2002
- Snes9x 2005
- Snes9x 2010
- SwanStation
- UAE
- VICE

For example, frame time deviations in a core like Snes9x 2010 are now extremely low with a default 64ms audio buffer. We measured 0.4 to 0.3% deviation, and this figure could likely be optimized even further by fiddling some more with audio buffer latency, or changing the audio driver.

Other measures have also been taken to further improve audio latency. Some cores have been updated now so that audio gets pushed to the frontend (i.e. RetroArch) AFTER the video frame has been uploaded. This is just in case the audio upload blocks for too long due to audio processing and syncing performed by the frontend. Uploading the video frame as soon as possible after the emulation loop is generally a good idea since it potentially avoids unnecessary input latency.

Steps have also been taken in cores to minimise use of the audio batch callback (for example, Tyrquake and Snes9x 2010), leading to better audio sample pacing, less overhead and better overall performance.

For Miyoo users

The following cores have been added for Miyoo platform users:

- BlueMSX
- ECWolf
- FCEUmm

For OpenDingux users

The following cores have been added for OpenDingux platform users:

- Dinothawr
- NXEngine
- XRick

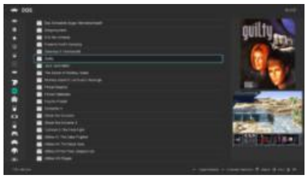
UI/UX improvements

RGUI, MaterialUI and Ozone menu drivers now have new color themes. 'Gray Dark / Gray Light' themes have been added.

For XMB users, vertical fade adjustments have been made so that it functions a bit better like the original. Title margin can now also be adjusted, so that the title won't cut off anymore on display devices with overscan (i.e. CRT TVs).

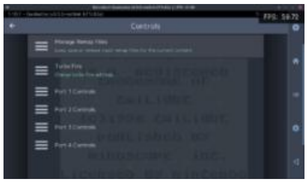
For RGUI users, 6×10 extended ASCII and Latin Extended A and B fonts have been added. These will enable most Latin alphabets to be displayed in RGUI.

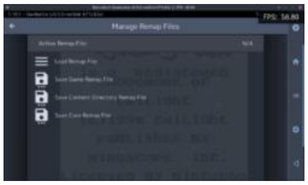
For Ozone users, a thumbnail scaling option has been added **(Settings->User interface->Appearance)**. This option scales the size of the thumbnail sidebar, which in turns means the thumbnails are scaled along with it. It should scale the thumbnails so that they should fill up more screen real estate now in the right sidebar. See the picture below for an example.



Manage Input Remapping Files

1.10.2 adds a new Manage Remap Files submenu to Quick Menu > Controls:





This updates correctly in real-time (the previous save/remove remap menu entries do not…), and only shows relevant options. When removing a remap, existing files are re-scanned and the one with the next highest priority (if found) will be loaded.

In addition, the currently active remap file will be saved automatically when closing content – i.e. it is no longer necessary (or indeed possible!) to save the file manually after each change.

We have also added a new Reset Input Mapping entry under Quick Menu > Controls > Manage Remap Files:

We have also fixed a nasty bug that could cause remap file corruption (incorrect or unwanted entries) when saving a remap after resetting one or more binds.

Prevent 'global' configuration of input libretro device type

Before, RetroArch allowed the input 'libretro device type' to be set globally per-user. This was nonsensical for a couple of reasons:

- Input device type has no meaning unless a core is actually running, so the option to set it globally does nothing but confuse users

- A global device type may be set that is incompatible with cores, leading to disabled input in-game. e.g. setting the global type to ‘RetroPad with Analog’ breaks Beetle PSX, since that is not a valid device type recognised by the core

1.10.2 fixes the issue by:

- Removing the global option Settings > Input > Port N Controls > Device Type (device type can now only be set while a core is running, via Quick Menu > Controls > Port N Controls)
- Ensuring that the ‘global’ device type is reset to the default RETRO_DEVICE_JOYPAD on startup and when closing cores (unless a device type override is set via the existing command line interface)

Note that device type is no longer stored in the main RetroArch config file, only in input remap files.

WiiU – Optimize for Gamepad

RetroArch WiiU adds a new option (Settings -> Video -> Output -> Optimize for GamePad). When enabled, it uses a 960p viewport if the user is on either 720p or 1080p (if they’re on 480p, they’re already optimized for GamePad). It defaults to off, so the native TV resolution is still preferred out of the box.

Explanation

The Wii U is a weird case with RetroArch because of the two screens (TV and Wii U GamePad). The Wii U can be configured to output video at 480p, 720p or 1080p (or interlaced equivalents), whereas the GamePad has a native 480-line display. While it is possible to send different images to the TV and GamePad, RetroArch currently sends the same image to both. This creates a bit of a conundrum as 480 does not divide evenly into any of the other available resolutions.

When running 240p content, setting the Wii U to 720p just works, because a 3x integer scale (240*3 = 720) on the TV also happens to be a 2x integer scale (240*2 = 480) on the GamePad. However, when running 480p content, having the Wii U set to 720p will result in a poor image all around, with non-integer scaling from 480->720 on the TV, and then even worse, 480->720->480 on the GamePad.

Running the Wii U at 1080p, you get the worst of all worlds. Absolutely nothing divides evenly into 1080, so no matter what content you’re playing, you’ll need either large borders or a filter/shader to stretch to that non-integer resolution, and putting that 1080p image back on the 480p GamePad makes it even worse again. Many users only use 720p because of the poor results you get from 1080p currently.

By running 1080p with a 960p viewport, you can do things like a 4x integer scale of 240p content which becomes a 2x integer scale on the GamePad, or a 2x scale of 480p content which becomes a 1x native display on the GamePad. Even more exotic resolutions like the Game Boy Advance (160p) are enhanced, with 160 dividing 6x into 960 or 3x into 480. This one change makes 1080p a lot less useless on Wii U.

- when “Optimize for Gamepad” is OFF: use the system output resolution (previous behavior)
- when “Optimize for Gamepad” is ON, pick the closest resolution based on the system resolution

And more

Not listed here of course are all the countless improvements made to individual cores since the last version. We might go into more detail on that sometime later, but rest be assured that cores are updated on a daily basis and receive heavy improvements, so keep updating your core library to get the latest benefits at all times!

Changelog

1.10.2

- 3DS: Add a menu toggle for switching between old and new 3DS speeds, located in the ‘Power Management’ menu. Enabled by default, hidden on old 3DS devices.
- AUDIO/MIXER: Free audio voices properly
- CHEEVOS: Update to rcheevos 10.3.3
- CHEEVOS: Support for Arduboy
- CHEEVOS: Fix tab sequences in rich presence being turned into t character
- CHEEVOS: Fix overflow when parsing float value that has more than 9 digits after the decimal
- CHEEVOS: Fix memory mapping when disconnect mask breaks a region into multiple blocks
- CORES: Enable manual selection of which cores are displayed in the ‘Standalone Cores’ menu
- DATABASE/EXPLORE: Added more categories to the Explore menu
- INPUT: Fix analog stick not working with ‘Unified Menu Controls’
- INPUT/MAPPING: Add ‘Manage Remap Files’ submenu + automatically save input remaps when closing content
- INPUT/MAPPING: Add ‘Reset Input Mapping’ option to ‘Manage Remap Files’ menu
- INPUT/MAPPING: Fix keyboard device remap nulling
- IOS/IOS13+: Support a toolbar that allows toggling of onscreen keyboard and touch mouse
- LIBRETRO: RETRO_ENVIRONMENT_SHUTDOWN fix – ensure core is properly unloaded when RETRO_ENVIRONMENT_SHUTDOWN is called
- LIBRETRO: RETRO_ENVIRONMENT_SHUTDOWN fix – ensure menu stack is properly flushed when RETRO_ENVIRONMENT_SHUTDOWN is called
- LINUX/MALI FBDEV: Fix segfault switching video threaded from quickmenu
- LOCALIZATION: Add Czech language support
- MMAP: Handle disconnect bits on both sides of len
- MIYOO: Improve CPU architecture and model name identification for Miyoo
- MENU/SETTINGS: Remove ‘Advanced Settings’ flag from ‘Settings > Core’ menu
- MENU/MATERIALUI: Add ‘Gray Dark + Light’ themes
- MENU/RGUI: Add 6×10 extended ASCII and Latin Extended A and B fonts. These will enable most Latin alphabets to be displayed in RGUI.
- MENU/RGUI: Add ‘Gray Dark + Light’ themes
- MENU/XMB: Add title margin adjustment

- MENU/XMB: Vertical fade corrections
- MENU/OZONE: The size of the thumbnail bar can now be changed though a new option (Settings->User interface->Appearance) up to double its normal size.
- MENU/OZONE: Add 'Gray Dark + Light' themes
- MENU/OZONE: Add thumbnail scale option
- HOTKEYS: Added hotkey for toggling sync to exact content framerate
- HOTKEYS: Prevent log spam when using rewind hotkey with cores that don't support rewind, if rewind functionality itself is disabled
- HOTKEYS: Add hotkey for toggling sync to exact content framerate
- PS3/PSL1GHT: Add RSX graphics support
- PS3/PSL1GHT: Add libco support
- PS3/PSL1GHT: Add experimental PSMove support
- RS90: Optimise layout of sdl_rs90_video
- STEAM: Use native OSK (Onscreen Keyboard) instead of built-in RetroArch version
- STEAM: New built-in core DLC downloader
- STEAM: Swap OK/Cancel buttons by default
- VIDEO/HDR: Removed redundant copy of buffer in HDR mode if the shader has already a HDR format i.e. R10G10B10A2 (updated Vulkan/D3D11/D3D12 drivers)
- VIDEO/HDR: Fixed crash when using stock shader and HDR and previous optimisation
- WAYLAND: Dynamically load libdecor at runtime
- WAYLAND: Fix splash screen when using xdg_toplevel
- WAYLAND: SHM anti-collision for the splash screen
- WAYLAND: Skip splash screen if window is not ready
- WII: Fix find_connection_entry(): needs unsigned int Otherwise the USB gamepad cannot be found, if VID/PID has leading zero. This issue happened with Retrode gamepad adapter
- WII: Rework Retrode gamepad implementation to support multi_pad interface
- WII: Fix – Unplugging and re-plugging now works again
- WII: vWii- Only gamepad 1 is supported, because multi_pad is currently only relevant in the Wii U implementation
- WIIU: Implemented the multi_pad interface according to input/connect/connect_wiiugca.c
- WIIU: Add Optimize for Gamepad option
- WIIU: Fix USB gamepad support

Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

Lakka 4.0 release (https://www.libretro.com/index.php/lakka-4-0-release/)

March 20, 2022 (https://www.libretro.com/index.php/lakka-4-0-release/)

Original article here (https://lakka.tv/articles/2022/03/19/lakka-4.0/).

New version of Lakka has been released!

We are happy to announce the new and updated version of Lakka.

Release summary

Changes since version 3.7:

- Build system based on LibreELEC 10.0.2
- RetroArch updated to 1.10.1
- Cores updated to their most recent versions
- superbroswar: added new libretro core
- sameduck: added new libretro core
- Mesa updated to 22.0.0
- Mainline kernel updated to 5.10.103 (PC, Amlogic, Allwinner, NXP)
- Raspberry kernel updated to 5.10.95
- Most ARM devices switched to aarch64
  Rockchip RK3288, RK3328 and RK3399 switched to mainline kernel 5.10.76
- Added support for additional Allwinner and Amlogic devices (not tested on our side, as we do not own many of these devices)
- Nintendo Switch: complete rewrite of the port with many fixes and enhancements
- Tinkerboard and MiQi now use common system RK3288 (hence the change of the image names); you need to place empty file named .nocompat in the /storage/.update folder / Update Samba share
- Dropped support for RPi4.arm

This is our first release with the updated build system. We tried to port all our changes to this new build system, but we might have missed some of our optimizations and enhancements. Please report us any regression or missing features or drivers via our issue tracker.

Not all platforms have been ported to the new build system yet. For these platforms we have released a maintenance update (3.7.1) with RetroArch and libretro cores updated to the same versions as in 4.0 release. Lakka users with Hardkernel's Odroid XU3/4, Odroid Go Advance / Super, and Anbernic's RG351M / RG351P / RG351MP / RG351V devices get update as well.

Known issues

See GitHub (https://github.com/libretro/Lakka-LibreELEC/issues) for information about currently open bugs and issues and also for possible workarounds for these bugs/issues.

Final notes

You can download the latest release from Lakka download page (https://lakka.tv/get/). If you want to follow the development of Lakka more closely, you can download latest Lakka nightly builds (https://nightly.builds.lakka.tv/latest/).

If you want to show your support for further development of the Libretro projects and ecosystem, you can learn more here (https://retroarch.com/index.php?page=donate).

Happy retro-gaming!

Lakka (https://www.libretro.com/index.php/category/lakka/), Libretro (https://www.libretro.com/index.php/category/libretro/), RetroArch (https://www.libretro.com/index.php/category/retroarch/)

Libretro/RetroArch New Cores – A5200/WASM4/Arduous/SameCDi/JumpnBump (https://www.libretro.com/index.php/libretroretroarch-new-cores/)

March 9, 2022 (https://www.libretro.com/index.php/libretroretroarch-new-cores/)

Libretro and its various contributors have certainly been busy these past few weeks. We bring to you a wide range of new cores, available for use in RetroArch, and/or any other Libretro-compatible frontend.

a5200

The a5200 core is an optimised Atari 5200 emulator based on Atari800 2.0.2. Originally developed for the GCW Zero, it runs full speed on even the weakest hardware (e.g. o3DS) – yet with a design focussed on ease of use and tight libretro integration it provides a simple 'plug and play' solution for Atari 5200 content on all platforms. The core debuts with robust input handling (including analog and number pad entry), audio filtering, CRT ghosting effects, an optional internal open-source BIOS and full save state support. Further enhancements are planned over the coming weeks.

WASM-4

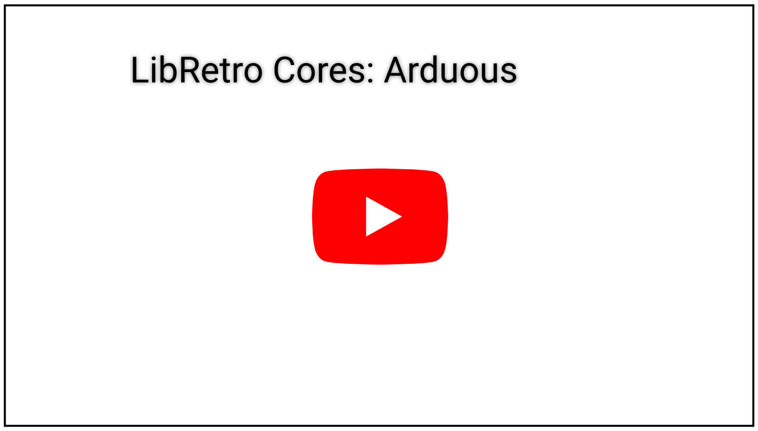
WASM-4 is a low-level fantasy game console for building small games with WebAssembly. Game cartridges (ROMs) are small, self-contained .wasm files that can be built with any programming language that compiles to #WebAssembly.

You can now play these games/programs from within RetroArch, or any other Libretro-compatible frontend.

You can find more information at here (https://docs.libretro.com/library/wasm-4/).

Arduous

A Libretro core for the #Arduboy. The Arduboy is a handheld game console with open source software, based on the Arduino hardware platform.



SAME_CDi

SONY DSC

SAME CDi is a S(ingle) A(rcade) M(achine) E(mulator) for libretro, forked from MAME libretro, which is in turn a fork of MAME. It includes only the Philips CD-i driver, and simplifies the loading of CD content to provide a ‘plug and play’ experience. It emulates exclusively the Philips CD-i game console/settop box device from the early ’90s.

You can find more information at here (https://docs.libretro.com/library/same_cdi/) and here (https://www.reddit.com/r/RetroArch/comments/t6ws9r/the_new_same_cdi_core_made_philips_cdi_emulation/?utm_medium=android_app&utm_source=share).

Jump ‘n Bump

Jump ‘n Bump has finally been ported to Libretro/RetroArch. This is a game for the whole family. You play as a cute fluffy little bunnies and hop on each other’s heads.

At the beginning you are in the menu, where you have to let each active player jump over the tree trunk to enter the play area, and then walk to the right. You will then enter the arena. The aim is to jump on the other bunnies’ heads…

Jump ‘n Bump was originally a DOS game by Brainchild Design, which was open sourced under the GPL license and ported to SDL, and then SDL2.

How to use it
You can play Jump ‘n Bump by loading one of the .DAT files (levels) from Load Content. A fair few of them are available on Libretro/RetroArch’s ‘Content Downloader’.

Arduous (https://www.libretro.com/index.php/category/arduous/), Philips CD-i (https://www.libretro.com/index.php/category/philips-cd-i/), RetroArch (https://www.libretro.com/index.php/category/retroarch/), Same_CDi (https://www.libretro.com/index.php/category/same_cdi/), WASM-4 (https://www.libretro.com/index.php/category/wasm-4/)

Older posts (https://www.libretro.com/index.php/page/2/)