ROCK N ROLLER

Offline reference for Rock N Roller — About, Features, Stems, Keybinds, and Settings. Live site: rocknroller.nicapotato.com.

About

A freeware guitar-practice app that plays Rocksmith-format `.psarc` charts from your own computer — your personal library stays on disk; the app never uploads it. Built around freeware CDLC from the CustomsForge community. Grant a local folder of your own CDLC to play. Optional curated demos load from our CDN only when you open Play with `?demo=songs` (charts) or `?demo=stems` (charts + stem mixes; larger download). How to generate your own stems and mix them in-game: stems & Demucs.

Trailer

> Watch trailer — youtube.com/watch?v=k--zDnyUZvs

What it does

Point it at a folder of CDLC `.psarc` files on your disk and it renders the charts on a 3D highway with audio, practice speed control (pitch-preserving time stretch), per-arrangement tuning info, playlists and profiles.

Your files stay on disk

Charts live on your machine. In the browser, the app asks for **read-only** access to a local folder via the folder picker; `.psarc` bytes are read in the page and never uploaded. Saved state (profiles, playlists, scan cache) lives in your browser's local storage for this site.

Chromium required (web)

The browser build needs a Chromium-based browser (Chrome or Edge). Folder access uses the File System Access API (`showDirectoryPicker`), which Firefox and Safari do not support. The build also runs multithreaded WebAssembly (Emscripten pthreads via `SharedArrayBuffer`), which needs a cross-origin-isolated page — something the site sets up automatically, but Chromium is still the supported target for the full library flow.

What it is not

This project is **not affiliated with Ubisoft or Rocksmith**, and it does not host, distribute, or bypass official Rocksmith DLC. It reads chart files you already have. Support official releases and the artists who make the music.

Platforms

Play in the browser at /latest (Chromium required), or grab desktop builds for macOS (Apple Silicon and Intel) and Windows from the versions page or itch.io. Desktop builds read `.psarc` folders from disk without the browser folder API.

Bugs, support, feedback

Join the Discord for bug reports and support, or to leave feedback and suggestions.

Features

How Rock N Roller solves its harder problems: local CDLC, catalog search, practice speed and section loops, Demucs stems, left-handed / inverted highway layout, and a 3D chart that stays locked to the audio — from one C + raylib codebase on macOS, Windows, and WebAssembly.

Stack & platforms

The app is C17 (plus one C++17 wrapper for Signalsmith Stretch) on raylib 6.0. One CMake project builds macOS (Apple Silicon and Intel), Windows x64, and Emscripten wasm from the same commit — platform differences stay in small shims for paths, dialogs, and threading.

Heavy lifting is a short list of libraries: SQLite for state and the song catalog, an in-repo PSARC reader, vgmstream for Wwise audio (no FFmpeg), ogg/vorbis, and Signalsmith for time stretch. The UI is drawn in-app; there is no separate toolkit.

Library that stays on your disk

Opening every `.psarc` on every launch would be slow. Instead, a local SQLite DB caches each song's metadata (title, artist, tunings, arrangements, length) keyed by file size and mtime. Startup loads the catalog from the DB; a background rescan only re-parses files that changed. Profiles, playlists, play history, and settings live in the same database. Charts are Rocksmith's binary SNG — parsed once when you pick a song, not from XML at runtime.

SQLite on disk (macOS / Windows)

On desktop, the database is a normal file on your machine — not in the cloud, not inside the app bundle. The state directory is created on first launch; the DB file is `rocknroller.db` (plus SQLite WAL sidecars while the app is open).

Platform	Path
macOS / Linux	`~/.nicapotato-app-saved-state/rocknroller/rocknroller.db`
Windows	`%LOCALAPPDATA%\nicapotato\rocknroller\rocknroller.db`
Web	Same logical path inside an IDBFS mount, persisted to the browser's IndexedDB for this site (see browser library access)

That file holds the song scan cache, profiles, playlists, play history, library folder list, stems folder path, autoplay mode, chart layout, and per-profile cell settings (including left-handed / invert). Your PSARC and stem audio stay wherever you put them — only metadata and prefs are in the DB. Reset in Settings wipes this state, not your media files.

Catalog search & query language

A large CDLC library is useless if you can only scroll it. The catalog search bar accepts plain text *or* a small query language that filters on the metadata already in SQLite — no reopening PSARCs.

Bare words are a global substring match (case-insensitive) across title, artist, album, year, and tuning. Column filters use `field="value"` and can be mixed with free text:

Clause	Filters on
`title="…"`	Song title
`artist="…"`	Artist
`album="…"`	Album
`year="…"`	Year
`tuning="…"`	Tuning (display label or raw offsets), e.g. `tuning="Drop D"`
`instruments="…"`	Arrangements present: `v` vocals, `l` lead, `r` rhythm, `b` bass. Optional per-arrangement tuning: `l-E Standard`, `b-Drop D`
`official="true|false"`	Official Rocksmith DLC vs CDLC
`stems="true|false"`	Songs with all six external stem MP3s indexed

Inside the quotes, **comma means AND** and `|` **means OR**. Examples:

- `stems="true" tuning="E Standard | Drop D"` — songs with stems whose tuning matches either string
- `artist="Rush" instruments="l, b"` — Rush charts that have both lead and bass
- `instruments="l-E Standard|b-Drop D"` — lead in E Standard *or* bass in Drop D

- `official="false" stairway` — CDLC only, plus free-text “stairway”

Clicking an album or artist card fills the bar with matching `artist="…"` / `album="…"` clauses. The same query powers the play queue when autoplay advances to the next song.

Browser library access

The browser has no filesystem, but the app needs a folder of your CDLC. The solution:

- Chromium's File System Access API grants **read-only** access to a local folder; the handle is remembered in IndexedDB so you don't re-pick every visit.
- The folder is mirrored into wasm MEMFS as **zero-byte skeleton files** with real names, sizes, and mtimes. Desktop scan/cache logic runs unchanged — and because the cache keys on size + mtime, cataloging never needs the file bytes.
- Opening a song stages the real PSARC on demand: a worker parks while the main thread copies bytes into a bounded LRU cache (512 MiB), then wakes. Large libraries stream through a fixed footprint.

Bytes go disk → page memory only. Nothing is uploaded. This (and folder picking) is why the web build needs Chrome or Edge.

Multithreaded wasm

The web build is the same C, compiled with Emscripten as multithreaded wasm: pthreads → Web Workers sharing memory via `SharedArrayBuffer`. That requires a **cross-origin-isolated** page (COOP/COEP headers). GitHub Pages can't send those, so a same-origin `coi-serviceworker` injects them on player URLs only (`/latest/` , `/0.1.35/` , … — one reload on first Play visit). Landing and docs stay un-isolated so embedded YouTube can load. The ~4 MB engine then streams from CDN; the site itself stays a thin shell.

Practice speed without pitch shift

Slowing audio naively drops pitch. Practice mode needs **0.3×–3.0×** with pitch held steady. Inside the audio callback, Signalsmith Stretch time-stretches each period: pull `rate ×` source frames, emit device-rate frames at original pitch. At 1.0× the stretcher is bypassed. The chart clock compensates for stretch latency so notes stay locked to what you hear; speed and seek reset the engine atomically.

Section loop

Rocksmith charts tag named **sections** (intro, verse, solo, …). Press **P** or the footer **Loop** button to loop the section under the playhead: the app seeks to that section's start and repeats until you turn loop off. While loop is on, **Enter** or **O** jumps back to the loop start and unpauses — useful after you scrub away mid-practice.

Loop edges show on the seek bar and can be dragged to tighten the span (as long as the chart has sections). Combined with pitch-preserving speed and stem mutes, this is the usual "drill this four bars" workflow. See keybinds.

Stems & Demucs MLX

Stem mode mixes six isolated tracks (drums, bass, other, vocals, guitar, piano) with per-stem faders. Separation runs offline in the companion Demucs MLX app (macOS Apple Silicon via MLX, or Windows x64 via NVIDIA CUDA), then Rock N Roller matches `{stems_root}/{psarc_basename}/` and streams the six MP3s. Full workflow, Mac vs Windows size/speed, folder layout, mixer, and `?demo=stems`: stems page.

Left-handed & string order

Two independent highway layout flags, saved per profile and per cell size class (also bulk-set from catalog Settings):

Setting	Default	What it does
String order (invert)	Red / low-E at **top**	Rocksmith-style default. **Inverted** puts red at the **bottom** of the lane — useful if you prefer that visual mapping.
Left-handed	Right-handed	Mirrors fret positions about the neck midpoint (Rocksmith-style lefty), so open/low frets sit on the opposite side of the highway.

Toggle them in Cell Settings while playing, or use Settings → "Set all to Left-handed / Right-handed" and "Set all to Default / Inverted" for every size class at once. Details: settings.

Long songs without blowing RAM

A long chart can be huge if kept fully renderer-ready. Songs over **10 minutes** keep the parsed chart in a compact cache and only expose a **3-minute window** to the highway. As you near the

edge (with preload that scales with practice speed), the window slides and refills. Shorter songs drop the cache after the first fill.

Audio streams similarly: OGG from disk, stems as MP3 with seek points, and long WEM-only tracks decode in the background while the chart starts. Memory stays flat for a three-minute song or a forty-minute one.

3D highway

Each instrument pane gets its own scissored 3D viewport (raylib `rlgl`) with an asymmetric projection so the camera can sit high without keystoning the frets. Camera modes track the active fret region on a 24-fret neck; zoom, runway grade, and lookahead are adjustable. Left-handed and invert flags remap frets and strings before draw (see above).

Notes are Rocksmith-style gems (halo / body / edge), with sustains as bendable ribbons and SDF fret labels on top. Each frame batches only the notes in the visible time window and draws them in a fixed depth order. Target is 60 FPS with four panes up.

Keeping the UI smooth

The main thread renders; workers handle library sync, song load, WEM decode, stem mixer open, and album art. They finish via atomic flags the frame loop polls — no blocking joins on the hot path. Speed, seek, and volume cross into the audio callback with atomics only, so the UI can't stall playback. The same model runs on web as the pthread pool above.

What ships

Desktop is roughly an **8 MB** binary; the web package is about **1.8 MB zipped** (~4 MB wasm + JS + a ~420 KB preload of fonts/shaders/logo — songs never ride along). That size is mostly “link only what Rocksmith content needs” and keep assets out of the download, not a separate size project.

Stems & Demucs

Stem mode lets you mix six isolated tracks — drums, bass, other, vocals, guitar, piano — each with its own fader (0–200%). Mute the guitar and play over the rest of the band, or solo a part to learn by ear. Combined with pitch-preserving speed and section loops, this is the usual "drill this four bars" workflow.

Separation does **not** run inside Rock N Roller. It is done offline with the companion Demucs MLX Audio Stemmer, then the game mixes what it finds.

> Demucs MLX on itch.io — nicapotato.itch.io/demucs-mlx-app

Companion app

Same raylib GUI on both platforms; the neural-net worker is what changes. Default model is Demucs `htdemucs_6s` (the six stems above). Input can be MP3, WAV, OGG, FLAC, M4A, or a Rocksmith `*_p.psarc` (it extracts the chart's audio first).

	macOS Apple Silicon	Windows x64
Backend	demucs-mlx (MLX + Metal)	Official Meta Demucs + PyTorch CUDA
Download	~132 MB	~2 GB (ships the CUDA runtime)
Speed	Far faster than realtime (about 73× on an M4 Max for 4-stem `htdemucs`). Unified memory; no extra GPU.	NVIDIA GPU recommended (~5–7 GB VRAM for 6-stem). CPU works but is slow (a 4-minute track can take about a minute). AMD/Intel GPUs fall back to CPU.
Needs	Apple Silicon Mac. Unsigned: `xattr -cr "DemucsMLX.app"`	Recent NVIDIA **driver** (not a CUDA toolkit). SmartScreen: More info → Run anyway.

The Windows zip is large because MLX does not run on Windows, so the worker bundles PyTorch's CUDA kernels. You do not install a CUDA toolkit. 3–4 GB cards may run out of VRAM on the default 6-stem segments.

Stemming workflow

1. In Demucs MLX, add an MP3/WAV *or* a Rocksmith `*_p.psarc`.
2. Pick an **output folder** — this becomes your Rock N Roller stems root.
3. Run separation. For `Eagles_Hotel-California_v1_1_p.psarc` it writes:

```
{stems_root}/Eagles_Hotel-California_v1_1_p/
  drums.mp3  bass.mp3  other.mp3
  vocals.mp3  guitar.mp3  piano.mp3
```

4. In Rock N Roller Settings → **Add STEMS folder**, point at that same `{stems_root}`, then resync. You can add up to 10 stems roots, same as PSARC folders. Details: settings.

How they link to PSARCs

Rock N Roller takes the PSARC filename without `.psarc`, looks for `{stems_root}/{that_name}/`, and requires **all six** MP3s. Only then is the song marked `has_stems` in SQLite. Paths are indexed per song; your PSARC and stem audio stay wherever you put them — the database only stores metadata and the stems-folder list.

Folder name must match the PSARC basename exactly. If the chart is `Foo_p.psarc`, the six files have to live in `Foo_p/`, not `Foo/` or a renamed folder.

Mixer in Rock N Roller

When a song has all six stem MP3s indexed, the **Stems** popup turns stem mode on and sets per-stem levels. Each fader runs **0–2.0** (shown as 0–200%; default 1.0) so you can mute or boost parts for practice. Playback opens six streaming MP3 decoders under one mixer that shares the same time-stretch path as normal audio, so practice speed still holds pitch. Long songs stream stems with seek points instead of loading them whole.

Mixer controls: settings. Transport and loop keys: keybinds.

Find songs that have stems

Catalog search filters on the SQLite flag, not by reopening files:

- `stems="true"` — songs with all six external stem MP3s indexed
- `stems="true" tuning="E Standard | Drop D"` — stemmed charts in either tuning

Full query language: features.

Try it on the web

`?demo=stems` loads curated demo PSARCs plus 6-stem MP3 trees into `/weblib/DemoStems`. That root is **additive** — it does not replace any stems folders you already granted. It is a larger download than `?demo=songs` (charts only). Chromium required; nothing is uploaded.

How the rest of the engine fits together: features. Desktop builds: versions.

Keybinds

Keyboard shortcuts for the song library and — most usefully — for practice inside a song. Mouse handles the rest (seek bar, volume, layout, stems, cell settings).

> **In song (cheat sheet):** `Space`/`K` pause · `J`/`L` ±10s · ←/→ speed presets · `P` section loop · `Enter`/`0` jump to loop start · `[`/`]` speed · `Z` zen mode · `Esc` back

Inside a song

These apply while a chart is open. Most transport keys are ignored while you're dragging a seek, loop, volume, or speed slider (or a cell-settings slider). Pause always works.

Key	Action
`Space or K`	Pause / resume
`J`	Seek backward 10 seconds
`L`	Seek forward 10 seconds
`← / →`	Step playback speed along presets 0.50 → 0.80 → 1.00 → 1.25 → 1.50 → 2.00 (clamp at ends; from a non-preset speed, snap to the next strictly lower/higher step)
`P`	Toggle **section loop** around the current song section (only if the chart has sections). Turning it on seeks to the loop start.
`Enter, keypad Enter, or 0`	Jump to the loop start and unpause (only when section loop is on)
`[`	Playback speed −0.1× (clamped 0.3–3.0)
`]`	Playback speed +0.1× (clamped 0.3–3.0)
`Z`	Toggle **zen mode**: hide the header and footer so the 3D highway fills the whole window (only the ZEN and Catalog buttons stay). Same as the ZEN header button.
`Esc`	Close an open play popup → else cancel speed typing → else return to the catalog

Typing a speed value

Click the speed number in the footer to type a rate directly (0.3–3.0). While that field is focused:

Key	Action
`Digits / .`	Edit the value (one decimal point)
`Backspace`	Delete last character
`Enter`	Commit if in range; otherwise discard focus
`Esc`	Revert to the current speed and unfocus

Footer buttons and sliders (previous / next / restart, Catalog, Loop, Auto-cycle, Vol, Speed, seek bar, loop edges) are mouse-only, as are cell header menus: Layout, arrangement source, viz mode, Stems, Cell Settings.

Catalog (song library)

Song table

Library nav, browse mode, search field not focused:

Key	Action
`↑ / ↓`	Move selection (hold to repeat; table auto-scrolls)
`Enter`	Load / play the hovered song
`Esc`	Dismiss nested UI in order: playlist edit → playlist popout → profile popout → else leave Albums / Artists / Settings back to Library

Playlist panel open

Key	Action
`↑ / ↓`	Scroll the playlist list
`Page Up / Page Down`	Scroll by one viewport

Search field focused

Key	Action
`Printable keys`	Insert at caret (or replace selection)
`Backspace / Delete`	Delete selection or character before/after caret
`← / →`	Move caret; Shift extends selection

Key	Action
`Home / End`	Jump to start/end; Shift extends selection
`Ctrl/Cmd+A / C / X / V`	Select all / copy / cut / paste (ASCII printable)
`Esc`	Clear search and unfocus

Albums / Artists filter

When that card's search is focused: type to filter, `Backspace` to delete, `Esc` to clear and unfocus.

Name / confirm modals

Key	Action
`Printable keys`	Type into the name field
`Backspace`	Delete last character
`Enter`	OK (empty name is rejected)
`Esc`	Cancel

Esc never quits the app — only the window close control does. See also settings and features.

Settings

What the different options mean — library folders and stems in the catalog Settings screen, autoplay, and the per-cell highway / lyrics controls while a song is open.

Library & stems

Open **Settings** in the catalog nav. This is where you point the app at your CDLC (and optional stem) folders.

Control	Meaning
PSARC folders	Up to **10** library roots. The first is the primary content folder; the rest are extras. Remove drops a root and resyncs the catalog.
Add PSARC folder	Pick a folder of `*_p.psarc` files. On the web this uses Chromium's folder picker (read-only).
Restore library (web)	Re-request access to a folder you granted before (browser permission).
Resync library	Rescan PSARCs (and stems). Unchanged files are skipped via the size/mtime cache; new or modified charts are re-parsed.
Add STEMS folder	Add a Demucs-style stems root (up to 10, like PSARC folders): `{folder}/{psarc_basename}/drums.mp3` … `piano.mp3` (six stems). Web mounts each under `/weblib/<folder name>`; `?demo=stems` adds `/weblib/DemoStems` without replacing user folders.
Reset	Wipe profiles, playlists, and the library cache after confirm. Destructive — your PSARC files on disk are not deleted.
Set all to Default / Inverted	Apply string orientation for all cell size classes: default = red/low-E at **top**; inverted = red at **bottom**. Same invert flag as Cell Settings.
Set all to Right-handed / Left-handed	Apply handedness for all cell size classes. Left-handed mirrors fret positions about the neck midpoint (Rocksmith-style lefty). Same `left_hand` flag as Cell Settings.

Autoplay

Cycled from the catalog Settings **Auto: …** button or the play footer **Auto** control. Controls what happens when a track ends.

Mode	Behavior
Auto: off	Stop at the end of the track
Auto: loop	Restart the current song
Auto: next	Play the next song in the filtered queue (default)
Auto: shuf	Pick a random next song
Auto: shuf*	Shuffle without immediately repeating the same song

Profiles & volume

The **Profile** button opens a popout to switch, create, or delete profiles. Each profile stores its own playlists, cell settings, and master volume.

Setting	Meaning
Game volume (Vol)	Master playback level, 0–1 (default 1). Adjusted with the play footer Vol slider; saved to the active profile when you release the slider.

While playing

Footer and cell-header controls that aren't on the catalog Settings page:

Control	Meaning
Speed	Practice tempo without changing pitch, **0.3×–3.0×** (default 1.0). Slider, typed field, footer **1x** reset, or `[` / `]` keys. See keybinds.
Loop	Section loop on/off — same as `P`. Needs chart sections; loops the section under the playhead. Drag loop edges on the seek bar when enabled. `Enter` / `0` jumps to loop start. See features § loop.
Transport	Previous / play-pause / restart / next, plus Catalog to leave the song.

Control	Meaning
Zen	Distraction-free mode — same as `Z`. Hides the header and footer so the 3D highway fills the whole window; only the floating **ZEN** and **CATALOG** buttons stay. Toggle the button (top right of the header, next to Catalog) or press `Z` again to bring the controls back. Great for portrait / mobile screens.

Cell Settings

Opened from a cell's **CELL SETTINGS** header. Values apply to that cell's **size class** (Quarter / Tall / Wide / Full) for the current profile — not just the one song.

Control	Meaning	Range	Default
Zoom	Camera distance / highway scale	0.5–2.5	1.0
Runway angle	How steep the highway grade feels (camera elevation)	0.35–2.2	1.0
Render distance	How many seconds of notes are drawn ahead of the strike line	1.5–6.0 s	3.0 s
Lyric size	Font scale for lyric strips / lyric cells	0.6–2.0	1.0
Lyric lines	How many lyric rows are visible	1–8	3
Inverted (red at bottom)	String order. Off: red / low-E at top (Rocksmith-style default). On: red at bottom.	off / on	off
Left-handed	Mirror frets about the neck midpoint for lefty players. Off = right-handed (default).	off / on	off
Reset (per row)	Restore that field's default	—	—

Quarter instrument cells expose zoom / angle / render / invert / left-handed. Lyrics-only cells expose lyric size and lines. Larger instrument cells expose the full set. Also see features § lefty.

Layout, source & viz

Control	Meaning
Layout	Chart grid template, e.g. Full; vertical/horizontal split; 4 Quarter; 3 highways + lyrics; tall/wide + quarter mixes. Optional **Auto-fill cells** assigns arrangements into empty slots. Saved with chart layout meta.

Control	Meaning
Cell source	What that pane shows: Empty, Lead, Rhythm, Bass, or Lyrics.
Cell viz	How the arrangement is drawn: **2D**, **3D**, or **3D + Lyrics**.

Stem mixer

When a song has all six stem MP3s indexed, the **Stems** popup can turn stem mode on and set per-stem levels (drums, bass, other, vocals, guitar, piano). Each fader runs **0–2.0** (default 1.0) so you can boost or mute parts for practice. Stem separation itself is offline (Demucs); the app only mixes what it finds. See stems & Demucs.

Keyboard shortcuts: keybinds. How the pieces fit together: features.