patchframe

Patchframe Demo Startup Guide

Patchframe is a visual instrument for people who make or perform music. Instead of treating visuals as a finished video or background decoration, it lets you build them from small connected modules, then shape them live while sound, movement, and timing change around you. This demo is the first hands-on version: a focused place to try the basic idea, make a simple patch, and feel how an image can become something playable.

Even with its focused module set, the Patchframe Demo is not just a preview or a mockup. It can be used as a real performance tool: you can build patches, connect modules, save and reopen your work, and use audio and MIDI input to make visuals respond to a live set. The demo keeps the instrument smaller and more predictable, but the core workflow is there: patch something, shape it, perform it, and bring it back later.

Basic Overview

Patchframe works like a modular visual rack:

1. Add visual modules to the canvas.
2. Connect source modules outputs into effects, mixers, and finally the **Output** module.
3. Use modulators such as **LFO**, **Audio Follow, Step Seq.**, or Random to move parameters over time.
4. Preview the result in the app, then send it to the **Output** window for performance or capture.

input ◎ output ▣ cv input ○ cv output ▢

The demo includes enough modules to build complete patches from still images, video files, generated shapes, lines, starfields, color correction, glitch, trails, transforms, mirrors, multiplication, simple mixing, audio following, sequencing, and oscilloscope-style monitoring.

Included Demo Modules

The demo is intentionally focused. Output resolution is locked to 1280 × 720, projection warp / mapping is off, and the module list is curated around the first patching experience.

SOLID COLOR
Creates a flat color source, useful as a background, mask, or simple signal.

SHAPE
Generates basic graphic forms that can be animated, layered, and transformed.

LINES
Draws line-based patterns for rhythm, grids, scanning movement, and structural visuals.

CELLULAR
Generates cellular procedural textures with organic, evolving detail.

STARFIELD
Creates moving point fields for depth, motion, and space-like visual energy.

IMAGE FILE
Loads a still image from disk into the patch.

VIDEO FILE
Plays a video file as a visual source.

TRANSFORM
Moves, scales, rotates, and positions an incoming image or texture.

CROP
Trims the visible area of an incoming texture.

COLOR ADJUST
Changes brightness, contrast, saturation, and color balance.

GLITCH
Adds digital breakup, distortion, and unstable signal character.

CHROMATIC TRAILS
Leaves color-separated motion trails behind moving images.

XMIRROR/KALEIDO
Mirrors and repeats an image into symmetrical visual patterns.

MULTIPLY
Combines images with a darkening blend, useful for texture and shadow layering.

MIXER 4
Mixes up to four visual inputs into one output.

LFO
Sends a repeating control signal for slow movement, pulsing, rotation, or modulation.

RANDOM
Produces drifting random values for less predictable motion.

STEP SEQUENCER
Creates stepped rhythmic modulation patterns.

AUDIO FOLLOW
Converts incoming audio level into a control signal for reactive visuals.

OSCILLOSCOPE
Displays audio or signal movement as a scope-style visual.

First Patch

Try this as the first five-minute patch:

1. Add **Video File** or **Image File** as a source.
2. Add **Transform** and connect the source texture into it.
3. Add **Mirror/Kaleido** or **Glitch** after **Transform**.
4. Add **Mixer 4**, connect the processed chain into one input, and connect the mixer to **Output**.
5. Add **LFO** and connect it to a movement, rotation, mix, or other effect amount parameter.
6. Let the patch run, then adjust parameters until the patch starts reacting like an instrument rather than a still composition.

For an audio-reactive version, add **Audio Follow**, allow microphone input when macOS asks, then map its cv output to any another visible parameter with cv input.