Peephole

A camera or capture card, full screen — in a browser tab, or as a desktop app that runs completely offline.

Using it

Open https://peephole.stoatworks-labs.com, press **Start**, and allow the camera when the browser asks. Then **Full screen**. Or install the desktop app, where it is the same three steps.

That is the whole tool. Everything below is detail you only need when something is not as you expect.

The bar along the bottom

Device	Every video input the browser can see — built-in cameras, USB cameras, and HDMI capture cards, which appear here like any other camera.
Mode	The resolution to ask the device for. *Whatever the device offers* lets the browser choose.
Fit / Fill	*Fit* shows the whole picture with bars where the shapes differ. *Fill* crops it to the screen.
Mirror	Flips left to right — what you want when someone is using it to look at themselves, and not what you want for reading anything on screen.
0° / 90° / …	Quarter turns, for a camera mounted on its side or a portrait screen.
Readout	What the device actually settled on, on the right.

Keyboard: **F** full screen, **M** mirror, **R** rotate, **C** fit or fill.

In full screen the bar and the mouse pointer fade after a couple of seconds of stillness, and come back as soon as you move.

Why the picture might be soft

This is the one that catches people, and it is not the page's doing: a capture card advertises a list of modes and the browser picks a conservative one, so a card carrying 1080p can open at 640 × 480 and look exactly like a bad cable.

Set **Mode** to the resolution you expect and watch the readout. If it says what you asked for, that is what you are getting. If it reports something smaller, the readout says so — *asked for 1920 × 1080* — which means the device refused, usually because the source is not really running at that mode.

The screen going dark

Where the browser supports it, Peephole holds a wake lock while a picture is up, so the screen does not dim or sleep. Firefox, and Safari before 16.4, have no such lock — the readout says so, and the machine's own energy settings are then the only thing keeping the screen on.

The desktop app has no such gap: it holds the display awake itself, on all three platforms, and keeps holding it when the window is behind something else.

Things it says, and what to do about them

It says	What is happening
The browser blocked access to the camera.	The site is not allowed to use the camera. The padlock in the address bar is where to change it, then press Start again.
Access to the camera was blocked. (in the app)	The operating system is refusing, not the app. It names the setting — **System Settings ▸ Privacy & Security ▸ Camera** on macOS, **Settings ▸ Privacy & security ▸ Camera** on Windows, the `video` group on Linux.
The device is there but would not open.	Something else has it — OBS, Teams, Zoom, another tab of this page. A camera can usually only be opened once.
The device stopped.	It was unplugged, or another application took it. Press Start.
That device is no longer there.	The remembered device is gone; pick another from the list.
Cameras are only available on a secure page.	You are on an `http://` address. Use the hosted page, or `localhost`. The app never sees this one.

Privacy

The picture goes from the device to the screen inside your browser. Nothing is recorded, nothing is uploaded, there is no account, and no server sees anything — the page is static files. Closing the tab ends it.

The only thing kept is your choice of device, mode and view, stored in this browser so the page comes back the way you left it.

What it deliberately does not do

No recording, no snapshots, no streaming out, and no audio. For production use OBS; to put a capture device on the network, see frame-ferret.

The desktop app

The same tool, installed, for a machine that has no route to the internet or should not be using one. Downloads are on the releases page.

Each platform has a build that runs on anything and a smaller one for a single processor type. Take the first column unless you know which machine you have.

	Runs on anything	Smaller
macOS	`macos-universal.dmg`	`macos-arm64.dmg`, Apple Silicon only
Windows	`windows-setup.exe`	`windows-x64-setup.exe`, or `-arm64-`
Linux	`linux-x86_64.AppImage` — download, make it executable, run	`.deb`, `.rpm`

The Windows `-portable.exe` files are the same thing without an installer, for running from a USB stick.

It is not a browser pointed at the website. There is no page being fetched and no local server; the whole application is inside the download, and it cannot reach the network even if you ask it to.

On macOS it just opens. The download is signed with an Apple Developer ID and notarised by Apple, so there is no warning to click through and nothing to run in Terminal. Drag it into Applications and open it.

On Windows, SmartScreen will warn you the first time, because the installer is not signed. *More info* then *Run anyway*.

The first Start asks for the camera, and that permission belongs to the operating system rather than to Peephole. If you refuse it, the app cannot ask again — the switch is in **System Settings ▸ Privacy & Security ▸ Camera** on macOS, and **Settings ▸ Privacy & security ▸ Camera** on Windows. On Linux there is no prompt; access depends on your user being able to read the device, which normally means being in the `video` group.

Differences from the tab, all of them small:

- Full screen is the window itself, so nothing of the app is left around the picture. **Esc** comes back out, as does **F**.
- The mode list always comes from the device, on every platform. In a browser that is true only in Chrome and Edge.
- The screen stays awake everywhere, with no caveat in the readout.
- There is no updater. A new version means downloading it, which is the point on a machine that is deliberately offline.