

webOSArchive

LunacyPublic

Notifications

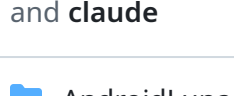
Fork0

Star6

main

Go to file

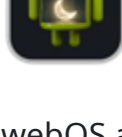
Code

 codepoet80 and claude	Device log: build 242...	0be49bc · 12 hours ago	
AndroidLuna	Db8: a property store...	12 hours ago	
CoreApps	Messaging 3.2.0: web...	3 days ago	
Docs	Device log: build 242 ...	12 hours ago	
LunaKeyboard	0.5.0: phone polish, M...	last week	
LunaRuntimes	Messaging's AIM on L...	3 days ago	
Meta	Add files via upload	2 weeks ago	
Workbench	Exchange security poli...	3 days ago	
gradle/wrapper	Lay the project out for...	3 weeks ago	
.gitignore	Signed release builds, ...	last week	
BUILDING.md	Bundled apps move to...	3 days ago	
CLAUDE.md	Bundled apps move to...	3 days ago	
README.md	Update README.md	4 days ago	
build.gradle.kts	Bundled apps move to...	3 days ago	
build.sh	Bundled apps move to...	3 days ago	
gradle.properties	Lay the project out for...	3 weeks ago	
gradlew	Lay the project out for...	3 weeks ago	
lint.xml	Lay the project out for...	3 weeks ago	
settings.gradle.kts	Lay the project out for...	3 weeks ago	

README

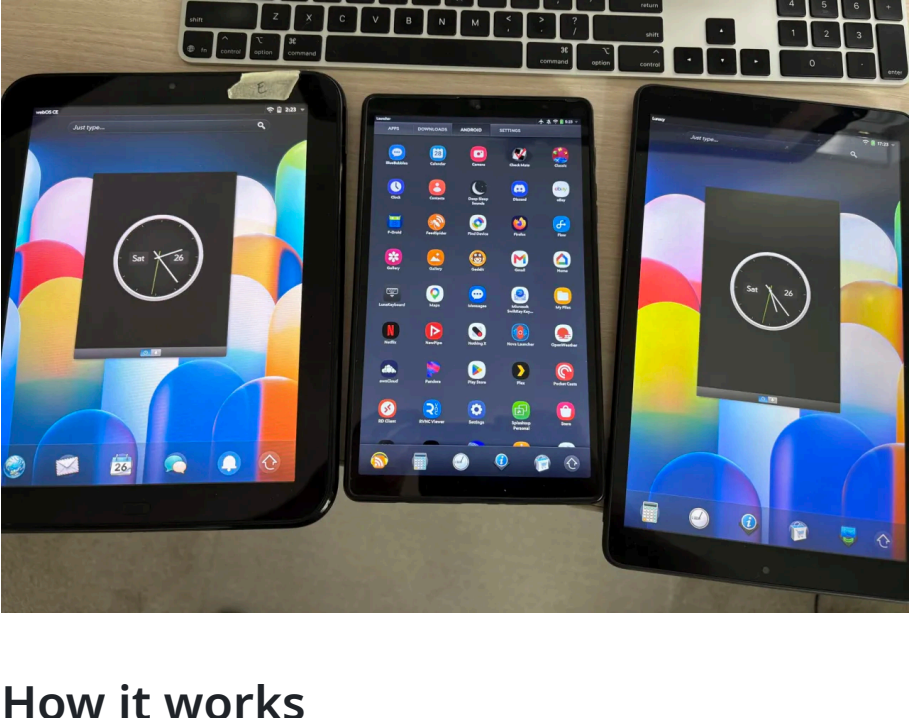
Lunacy

Run legacy webOS apps on modern Android phones and tablets, inside a shell that looks and feels like webOS.



webOS apps were HTML and JavaScript. They loaded their framework (Enyo or Mojo) from the OS, ran as cards in the Luna system shell, and talked to the system over the Luna service bus (`pa1m://...`). The devices are more than a decade old, but the apps are still good. Most of them never needed Palm's hardware. They needed Palm's *contract*.

Lunacy provides that contract on Android.



How it works

Lunacy re-implements the contract, not the OS:

- A simulated Luna shell.** A native Android UI that recreates the webOS experience: card view, gestures, launcher, notification banners and dashboard. None of Palm's code runs. It looks and behaves like webOS, and it has to feel right.
- Modernized frameworks.** Apps load their framework from `/usr/pa1m/frameworks/...` and trust the OS to provide it. Lunacy supplies the TouchPad's own [Enyo 1.0](#) (Apache 2.0), fixed to run on a modern Chromium renderer. Fixes in the library cover every app built on it.
- Apps as cards.** Each app runs unmodified in its own WebView card, on its own origin.
- A simulated service bus.** `pa1m://` calls are routed either to Lunacy itself (app manager, db8, shell services) or to the equivalent Android feature (connectivity, power, locale, notifications, intents).

Enyo was designed to scale from phone to tablet: its flex layouts and panels fill whatever screen they are given. So most apps run at the device's real size, on both phones and tablets.

What comes first

The look and feel is the TouchPad's, measured on an original device: how a card opens, how the launcher moves, what a loading card does, where every pixel of the shell sits. Fidelity is the project's first rule, ahead of speed or simplicity, and every difference the engine forces is measured and written down ([Docs/luna-deltas.md](#)).

What it is not

- Not an emulator.** No Palm binaries, no system image, no QEMU.
- Not a per-app porting kit.** No app is patched. Every fix goes into the frameworks or a global compatibility layer that treats all apps alike.
- Not LuneOS.** [LuneOS](#) is a full OS. Lunacy is an app on the Android you already have.

Scope and order

- Enyo 1 apps (webOS 3.x, TouchPad era), and Enyo 2 apps, which bundle their own framework can run on any OS, but feel at home on Lunacy.
- Mojo apps (webOS 1.x–2.x)
- JS (Node) services that ship with apps
- PDK native apps
- Hybrid apps: web apps that interact with a PDK plug-in

Apps come from webOS Archive's [Museum](#), through the TouchPad's own App Catalog. It is itself an Enyo app, so Lunacy bundles it and it runs inside Lunacy like any other card. The shell's styling target is LunaCE.

Platform

Lunacy supports Android 5.0.1 through Android 17 on tablets, as a 32-bit APK for the older devices and a 64-bit one for the newer. Phones are partially supported through light adaptation or zooming.

- Can run as the default Android Launcher (optional)
- Pair it with the optional webOS-style keyboard
- Launches webOS and Android apps

Using it

Lunacy is sideloaded: there is no store listing for now.

- Allow unknown sources.** Settings → Security → Unknown sources. (On Android 8 and later this is asked for per app, when you open the APK.)
- Install the APK for your device** - `AndroidLuna-arm64.apk` for a 64-bit device, `AndroidLuna-arm32.apk` for a 32-bit one (every Android 5 device; if unsure, try the 64-bit one first, which a 32-bit device refuses). Copy it to the device and tap it, or `adb install` . Open **Lunacy** from the Android launcher. The card view, the launcher and the bundled apps are there; **App Catalog**, on the launcher's Downloads tab, is where the apps come from.
- Optional: the webOS keyboard.** Install `LunaKeyboard.apk` , open **LunaKeyboard** and use its two buttons - switch it on in Android's settings, then choose it while a text field has focus. Android's own keyboard stays installed; switching back is the same picker.
- Optional: Exhibition as the screen saver.** Settings → Display → Daydream, turn it on and pick **Lunacy Exhibition**. Whenever Android would sleep, Lunacy shows webOS's Exhibition mode instead. Which face it shows is chosen in Lunacy's own Exhibition app, on the Settings tab. Lunacy never turns Daydream on by itself.

Building from source is [BUILDING.md](#).

Documentation

- [BUILDING.md](#): what you need and how to build both APKs
- [Docs/architecture.md](#): components and boundaries
- [Docs/roadmap.md](#): phases and exit criteria
- [Docs/fix-log.md](#): every compatibility fix, the layer it landed in, and how each app fares
- [Docs/db8.md](#): webOS's database, and what Lunacy answers for it

About

webOS application environment for Android

[android](#) [enyo](#) [mojo](#)

[webos](#)

Readme

Activity

Custom properties

6 stars

0 watching

0 forks

Report repository

Releases

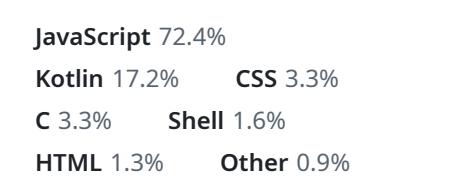
10 tags

Contributors 2

claude Claude

codepoet80 codepoet

Languages



- [Docs/lessons.md](#): the rules this and earlier attempts have paid for
- [Docs/mojo.md](#): how Mojo is packaged, and what it takes to run it
- [Docs/spike-1.md](#): stock Enyo on an Android 5 tablet, first results
- [Docs/android5-setup.md](#): what is done to an Android 5 device
- [Docs/dev-workflow.md](#): building, running, and checking against a TouchPad
- [Docs/luna-shell-reference.md](#): LunaCE shell spec, from its source
- [Docs/luna-deltas.md](#): where Lunacy still differs from LunaCE, as a work list with pointers
- [Docs/related-projects.md](#): the wider webOS preservation landscape

Companion apps

- [LunaKeyboard/](#): an optional Android input method that looks and behaves like the TouchPad's keyboard, scroll ball and all. A separate APK that shares nothing with Lunacy and works in any Android app.

A project of [webOS Archive](#).