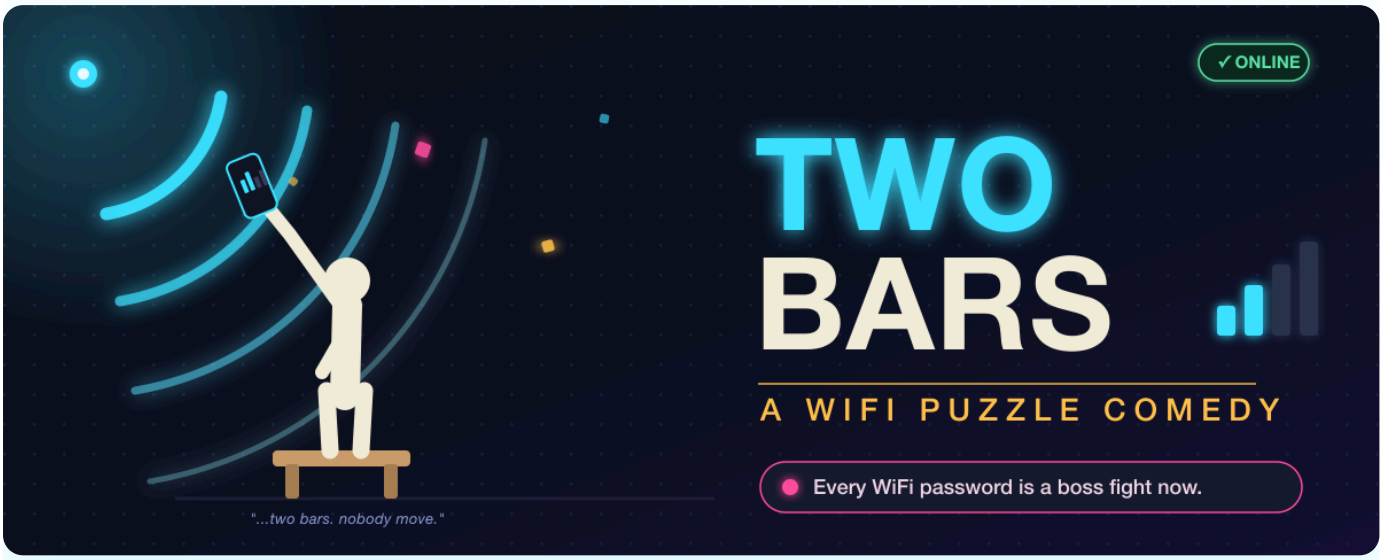
TWO BARS

A WiFi Puzzle Comedy — Investor Brief & Player Manual



> *"You've entered the password. It's wrong. It has always been wrong. Welcome to WiFi."*

Part I — The Pitch

The one-liner

Two Bars is a cozy-frustration puzzle-adventure about the last great human struggle: **getting onto the guest WiFi.** You play a mildly competent adult whose whole life migrated online the moment their connection died — and every locked network is a self-contained comedy mystery you crack **not by hacking, but by detective work.** It was always the dog.

The fantasy

In this world, WiFi is *visible* — shimmering signal arcs, a router humming in the corner, dead-zone fog where the bars die. You are the one person who can read a room's signal like a hunter reads tracks: find the sticky note under the router, charm the passive-aggressive barista, out-bureaucrat the captive portal, and deliver that specific, universal, dumb triumph — **finally getting online.**

Why it works

UNIVERSAL, EVERGREEN PREMISE	Everyone alive has stood in a kitchen holding their phone aloft, whispering "*nobody move.*" It has never been made into a game.
A GENUINELY NOVEL CORE MECHANIC	The password is *never* guessable cold. You assemble it from clues found in the world — *Obra Dinn* deduction, not brute force. No competitor has this.
IT MARKETS ITSELF	The comedy is recognition-based. Every cursed network name and 4-second airport-WiFi speed-run is a screenshot someone *wants* to post.

APP-STORE CLEAN BY CONSTRUCTION	The solution is always a sticky note or a conversation — never a command. Zero real networking capability, zero legal risk.
SMALL-TEAM-SHAPED	Flat-vector art, a lean hand-rolled engine, data-driven levels. New levels are *content, not code.*

The market & model

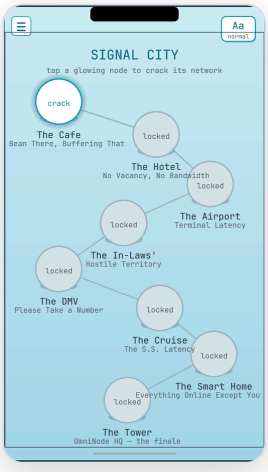
Premium narrative puzzle games — *Monument Valley, Untitled Goose Game, Gorogoa, Return of the Obra Dinn* — prove that one clever mechanic wrapped in charm sells millions at a healthy price with no ads. Two Bars sits in that lineage with a hook that is **more universally relatable** than any of them. Model: free first chapter, then a single one-time unlock (**premium, ~$4.99–6.99**). No ads, no energy timers, no dark patterns. iOS-first, then desktop.

Where the build is today — this is real, not a mockup

This is **not a slide deck of concept art.** There is a working, tested, playable build:

- ✓ **8 complete, playable levels** — a full campaign, The Cafe through the OmniNode Tower finale
- ✓ **The entire core loop** end-to-end: explore → gather clues → deduce → crack → connect
- ✓ **4 speaking characters** with branching dialogue
- ✓ **Full accessibility** — adjustable text size, reduce-motion, colour-blind support, a "fair" mode, all reachable from an in-game ≡ Settings menu, plus first-run coaching that hand-holds new players through every level
- ✓ Built on **raylib 6 in C**, structured so the **iOS port is a shell, not a rewrite**
- ✓ **115 automated checks** — every level provably winnable, the whole campaign machine-verified

The screenshots in this document are from that live build.



Signal City — the campaign map. Eight networks, each a comedy mystery; crack one to unlock the next.

The ask

We are raising to take this validated vertical slice to a **polished, full-content App Store launch**: art production, audio, the remaining signature levels, marketing, and platform certification. The mechanic is proven and the engine is built. What remains is content and polish — **the most predictable, least risky phase of game development.**

Part II — The Player Manual

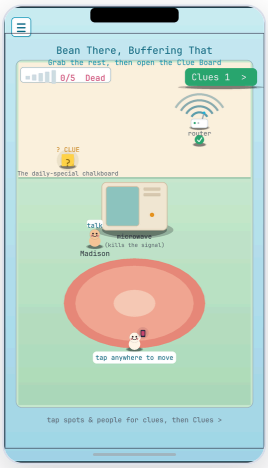
What you are doing

Every level is one locked WiFi network. Your job is to figure out the password — like a detective, not a hacker. Nobody types `sudo` in this game. You find clues, talk to people, notice the café's founding year chalked on the wall, and *deduce.*

The Connection Cycle — the loop of every level

```
SPOT (map)  →  SCOUT (room)  →  GATHER (clues)  →  CRACK (puzzle)  →
                        CONNECT (the ✓)
```

1. SPOT	On the **city map**, tap a glowing node. Locked nodes are dim — crack the one before it first.
2. SCOUT	You're dropped into an **isometric room**. Walk by tapping. Your **signal bars** rise near the router and drop behind obstacles.
3. GATHER	Tap the glowing **amber spots** for clues — a sticky note, a chalkboard, a keycard. Tap **people** to charm a clue out of them.
4. CRACK	Open your **Clue Board**, lace the clues into a candidate, and hit **Crack it** to play **Passwordle**.
5. CONNECT	Get it right and the world erupts: bars slam full, the room warms, the modem handshake resolves, the sacred ✓ lands. **The next network unlocks.**



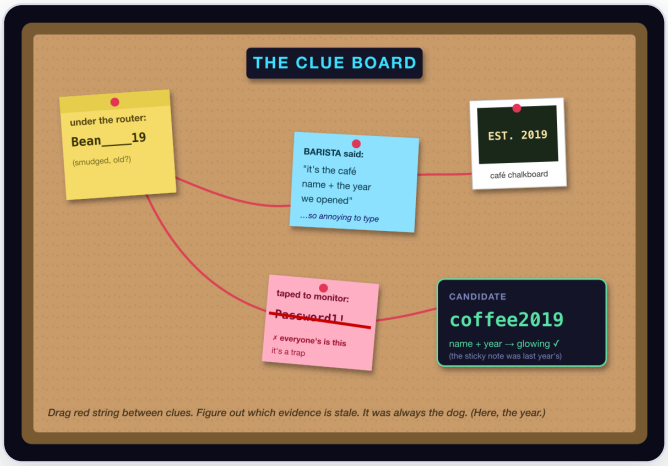
Inside a level: the router, signal arcs, clue hotspots, an NPC, and your signal bars reading the room.

The mini-games

🔍 **Signal Hunt** — Your bars update live as you move: strong near the router, dead in the corners. **Some evidence will not show itself until your signal is strong enough** — you can see there is something on that counter, you just cannot read it from the doorway. *(Signal shows as bars **and** a label — "2/5 · Weak" — never colour alone.)*

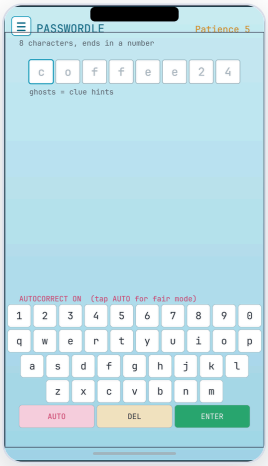
🧩 **The Clue Board** — Every clue is a typed token: a `WORD`, a `NUMBER`, a `NAME`, a `DATE`. Nothing arranges itself: **you** choose which fragments to lace and in what order, and that order *is* the password you are about to try. **The twist:** some clues are *stale* — the faded sign is the year the *previous* owners opened; the framed

photo is the *previous* dog — and a stale clue looks exactly like a good one until you have spent a guess on it. Then the board tells you why it lied, and remembers. Spotting which evidence to *ignore* is the game.



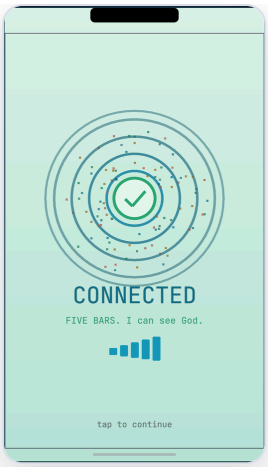
The Clue Board. Combine fragments; work out which evidence is stale. It was always the dog — here, the year.

⌨️ **Passwordle** — A Wordle × Mastermind hybrid. You type the candidate **you** assembled — the game never pre-fills it. Letters you have *proved* correct linger as faint "ghosts" for the next attempt. Meanwhile the on-screen keyboard has an **autocorrect gremlin** that capitalises your first letter, exactly like your real phone. Wrong guesses spend **patience**, and running out ends nothing — you are sent back to the room to think again, evidence intact. Frustrated? One tap flips **Fair Mode** and the sabotage stops. The joke is never a wall.



Passwordle in the live build. You type what you deduced; confirmed letters linger as ghosts; the keyboard is quietly out to get you.

💬 **Sweet-Talk** — Some passwords live in a person's head. Tap an NPC and *fish* for the clue. **RAPPORT** warms them, **PROBE** goes for the info (risking suspicion), **PIVOT** cools things down. Push too hard and they stop answering — *"You know what? No."* — and you must walk away, let them cool off, and come back better. It costs time, never the level. Read the person: the barista melts for latte-art flattery; Chad from IT just wants someone to complain about users with.



The money shot. Over-invested on purpose: the ✓ is the most satisfying button in the game.

The eight networks of Signal City

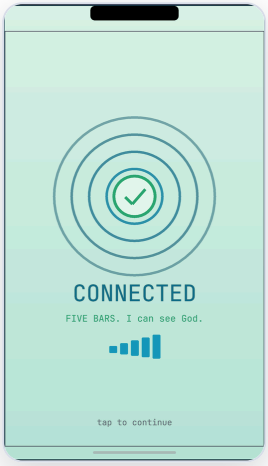
#	LEVEL	THE PREMISE	THE COMEDY
1	**The Cafe**	Free WiFi "with purchase." The barista weaponises the terms.	Password is `coffee` + the founding year — but the special is spelled "ØΔtMīlk LΔtté."
2	**The Hotel**	Per-device, per-day WiFi. The card code doesn't work.	An old keycard in the drawer is a beautiful, useless red herring.
3	**The Airport**	30 free minutes, then a paywall. Four decoy networks.	The gate code is on your boarding pass; the bin has yesterday's cancelled flight.
4	**The In-Laws'**	Sunday dinner. Router behind the good-china cabinet.	The password is the dog's name — but there've been several dogs. Gerald helps if you compliment Rex.
5	**The DMV**	Bureaucratic purgatory. A 47-page laminated EULA.	Network `GUEST`, password `guest`. The hardest level, because you refuse to believe it.
6	**The Cruise**	Satellite internet at highway-robbery prices.	Ship name + deck number — and a keycard from a different ship, a happier time.
7	**The Smart Home**	The fridge, doorbell, and toilet are online. You are not.	The Doorbell has admin rights, somehow, and blurts the port number if you appreciate its work.
8	**The Tower**	OmniNode HQ. The source of all throttling. The finale.	Chad from IT keeps closing your ticket as "resolved." Befriend him for the CEO's ego-password.

Controls & accessibility

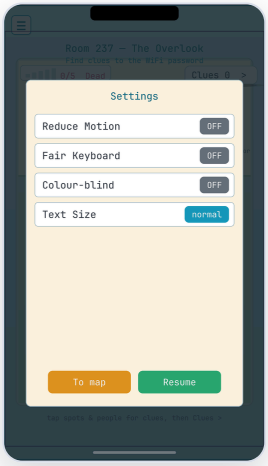
Designed **touch-first** for phones, fully playable with a mouse on desktop. **Tap** to move, collect, choose, and press keys; **type** your guess in Passwordle; the ≡ button opens the Settings/pause overlay (change options, jump To map, or Resume); **Back/Esc** to leave a menu. Accessibility is built in from day one:

- **Settings anywhere** — a ≡ menu on the map, room, Passwordle and Sweet-Talk opens a pause/Settings overlay with one-tap toggles for Reduce Motion, Fair Keyboard, Colour-blind and Text Size (Normal/Large/Huge), plus To map and Resume. Every accessibility option is fully touch-reachable on a phone — no keyboard needed.

- **Text Size** — an **Aa** button on the map (or the Settings menu) cycles Normal → Large → Huge; every screen reflows so nothing clips, and the choice is remembered. Legible at arm's length, and the web build renders crisp on retina screens via a HiDPI backing buffer.
- **First-run coaching** — a bouncing pointer and tip banner hand-hold new players through *every* level until it's cracked (next note → the person hiding a clue → the board → the answer), retiring level-by-level as each network falls.
- **Reduce Motion** — removes screen shake, the particle fountain, and the keyboard wobble.
- **Fair Mode** — disables the autocorrect sabotage entirely.
- **Colour-blind support** — and by design the game *never* relies on colour alone: signal carries a number and label, clues carry a type tag.



The same victory in Reduce-Motion: all the catharsis, none of the shake or particles.



Every accessibility option, one tap away on a phone — the ≡ Settings overlay.

Part III — Under the Hood

Two Bars is built in **C on raylib 6**, macOS-first, structured for a clean iOS App Store port.

- **Everything is data.** Levels, rooms, puzzles, dialogue, and every line of text are JSON loaded at runtime. The eight-level campaign was built *without touching the engine.*
- **Fully machine-verified.** A headless harness drives the real binary with scripted input and inspects a complete JSON state log of every frame. **115 automated checks** confirm every level is winnable, that clues never lace themselves, that lacing order changes the answer, that a signal-gated clue really is

unreachable from a dead zone, that stale evidence explains itself, that running out of patience never soft-locks, and that the whole 8-level campaign completes — with no display, on every change.

- **Plays in the browser — and remembers.** The full game compiles to WebAssembly and runs with no install, crisp on retina phones; progress now persists across reloads, with saves that migrate forward across version bumps and self-recover from a backup.
- **Honest, portable architecture.** A 120 Hz fixed-step loop, a scene stack, one semantic input layer that already takes both a mouse and a touchscreen, and a platform seam isolating every OS-specific line for iOS.

> ***The bottom line:*** *the risky, unproven part of a game — is the core loop actually fun, and does it hold together? — is already built, playable, and tested. What remains is the predictable work of content and polish.*

Two Bars — a WiFi puzzle comedy. The "hacking" is fictional, social-deduction puzzle flavour; the game has zero real networking capability. Screenshots are from the live development build. Repo codename `wifi-hacker`; ships as **Two Bars**.