When Worlds Collide: How *Flymoon* Turns an Accidental Idea into Automated Astronomy

A new open-source tool quietly watches the sky — and catches every plane that dares cross the face of the Sun or Moon

Feature · February 2026

Flymoon was originated and prototyped by David Betancourt Montellano. The concept has since been expanded by open-source contributors. The code is freely available at https://github.com/dbetm/flymoon.

There are moments in amateur astronomy that seem almost orchestrated by the universe itself: a jet slices silently across the face of a full Moon; a commercial airliner becomes a black dot skating over the solar disk. Most of us miss them. They last, at most, a second. Without forewarning, without your camera rolling, they vanish as completely as if they never happened.

The question at the heart of Flymoon is simple: what if you never had to miss one again?

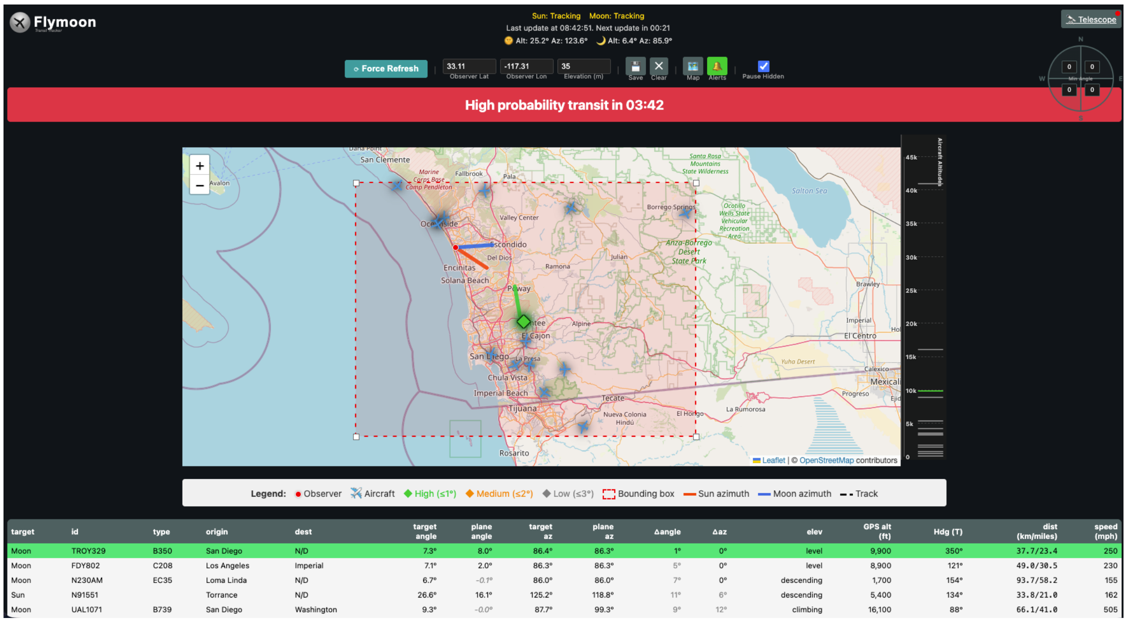
The Idea: Predict, Don't React

The premise behind Flymoon is elegant. Commercial aircraft broadcast their position, speed, altitude, and heading continuously via ADS-B transponders — the same signals that populate apps like FlightAware and FlightRadar24. A solar or lunar telescope, meanwhile, is pointed at a target whose position in the sky is precisely calculable. If you know where every aircraft *is* at every moment, and you know exactly where your target *is* in the sky, you can compute which flights will cross the disk — and *when*.

Flymoon does exactly this in real time, refreshing flight data from FlightAware's AeroAPI on an adaptive schedule and continuously recalculating transit predictions. When a high-probability transit is detected — defined as an angular separation of less than one degree from the target center — the system sounds an alert, displays a countdown, and (if enabled) automatically triggers video recording on a connected ZWO Seestar smart telescope.

The recording window is precise: video begins ten seconds before the predicted transit and ends ten seconds after. If a second flight is predicted to cross within that window, the recording simply extends to cover it. Nothing is missed.

The Flight Map: Situational Awareness at a Glance



Flymoon flight tracker map showing aircraft with transit predictions

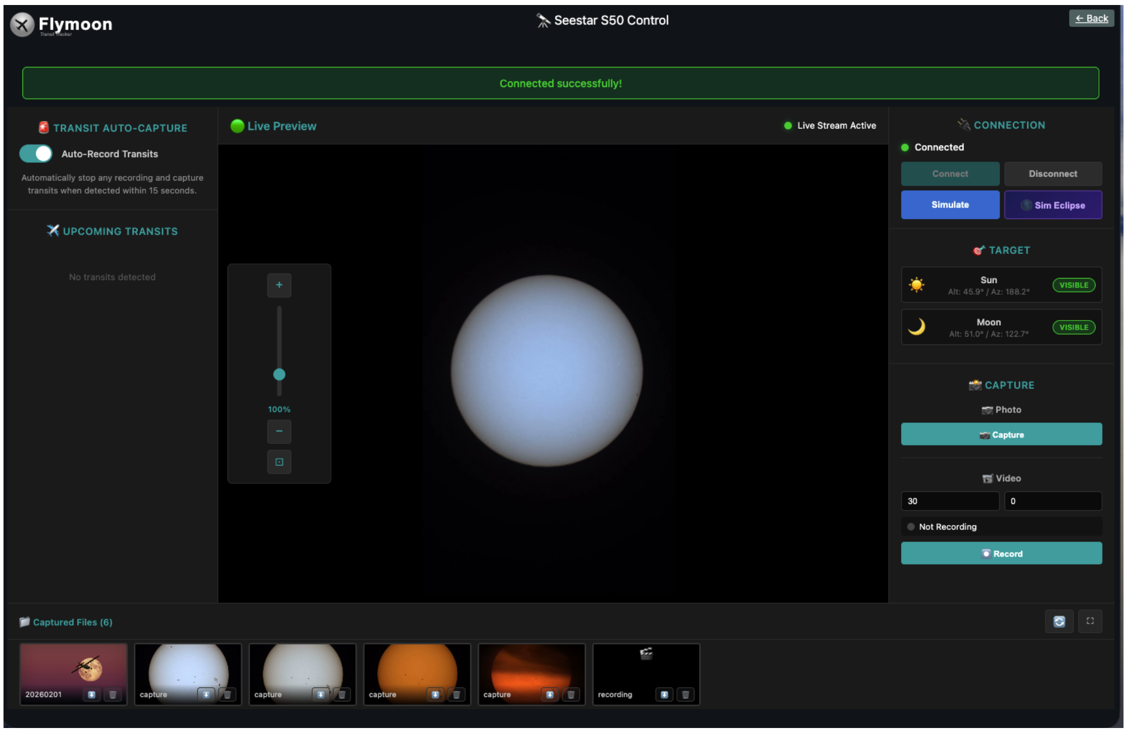
Figure 1 — The Flymoon main interface showing live ADS-B flight data. Aircraft are color-coded by transit probability: green diamonds (≤1°), orange squares (≤2°), grey markers (≤3°). The red dashed bounding box defines the observable sky region. The orange line points toward the Sun; the blue dashed line toward the Moon. A high-probability transit is flagged in the red banner — 3 minutes 42 seconds away.

Open Flymoon and the first thing you see is a live Leaflet.js flight map. Every aircraft in the configurable bounding region is plotted and color-coded by transit probability. Green markers signal a potential disk-crossing at less than one degree of angular separation — high probability. Orange means within two degrees. Azimuth arrows radiate from the observer's location toward the Sun and Moon.

Below the map, a sortable table lists every tracked aircraft with its target angle, plane angle, angular separation, altitude, speed, and predicted transit time in minutes. Click any aircraft on the map — or click its row in the flight table — and Flymoon pulls up the full FlightAware historical track along with the computer-projected forward path, drawn directly on the map. You can see exactly where that aircraft came from and precisely where it is headed relative to your target. Click again to release the selection and return to the full overview.

A countdown banner at the top of the page — red for high probability, amber for medium — ticks down to the moment of transit in real time.

The Telescope Interface: Connected and Intelligent



Flymoon Seestar control interface showing live lunar preview and connected status

Figure 2 — The Seestar control panel with a live MJPEG preview stream showing the Sun. The telescope is connected (green indicator), both Sun and Moon are VISIBLE in the current session, and the filmstrip at the bottom shows six captured files from previous transits.

Flymoon's telescope control panel is where the automation comes to life. After connecting to the Seestar over the local network, the interface streams a live MJPEG preview directly from the telescope. The user can see exactly what the scope sees, zoom in via the in-app slider, capture still images, or start video recordings — all without leaving the browser.

Flymoon works with both the ZWO Seestar S50 and the S30, communicating directly over the local network using the same JSON-RPC protocol the telescope's own app uses — no additional bridge software required.

With "Auto-Record Transits" enabled, Flymoon watches every incoming transit prediction and triggers recording automatically. When a transit is imminent, a compact status banner appears at the bottom of the preview image — a slim red bar reading "🎯 TRANSIT CAPTURE IN PROGRESS" — unobtrusive enough that the live view remains fully visible.

The system handles overlapping transits intelligently. Rather than interrupting one recording to start another, it simply extends the end time. If three aircraft are predicted within twenty seconds of each other, the recording runs continuously through all three.

A filmstrip at the bottom of the page accumulates every captured file from the session: still photographs, video recordings, and simulation captures, each with a thumbnail, filename, and one-click playback.

Eclipse Awareness: Thinking Like a Forecaster

Perhaps the most ambitious feature in Flymoon is its eclipse awareness system — an NWS-style multi-level alert architecture that treats eclipses with the same methodical seriousness that meteorologists apply to severe weather.

An eclipse isn't just a single event — it's a process. First contact, totality, last contact. An aircraft transit *during* an eclipse is an incredibly rare thing. Flymoon wants you ready long before C1, recording before it starts. And if a plane crosses during totality? That's once-in-a-lifetime.

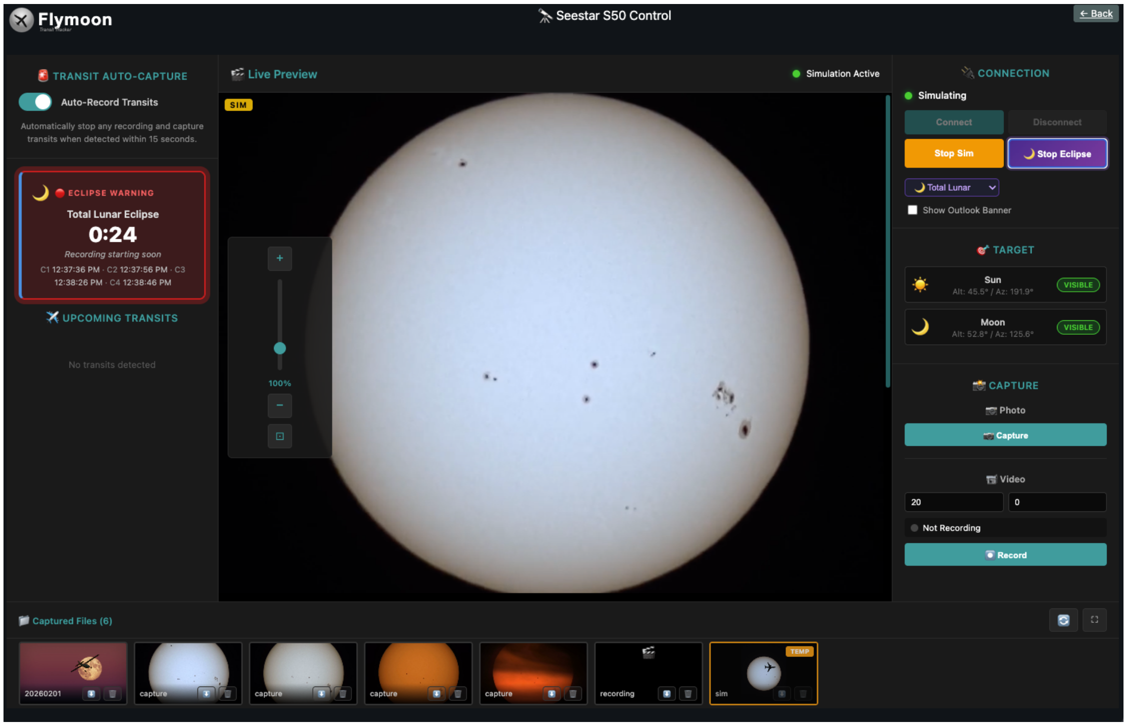
The system issues alerts across five escalating levels:

Level	Trigger	Response
Outlook	Eclipse within 48 hours	Banner notification only
Watch	Eclipse within 60 minutes	Eclipse card with countdown
Warning	30 seconds to C1	Pulsing alert; recording arms
Active	C1 through C4	Eclipse card; recording pinned to C4+10s
Cleared	Post-C4, 30-minute window	Summary card; file saved to filmstrip

Eclipse detection runs entirely in Python on the backend, using Skyfield for precise ephemeris calculations. For lunar eclipses, contact times are derived from the umbral geometry. Solar eclipses are detected by scanning the six-hour window around each new moon and using binary search to find C1 and C4 to within ten seconds.

When a solar or lunar eclipse is Active and an aircraft transit occurs simultaneously, Flymoon doesn't start a new recording — it extends the existing eclipse recording and drops a ✈️ marker in the filmstrip thumbnail, a permanent record of one of the rarest conjunctions in observational astronomy.

The Eclipse Simulator: See It Before You Need It



Flymoon simulator mode showing a Total Lunar Eclipse Warning card with 24-second countdown

Figure 3 — The eclipse simulator in Warning phase, 24 seconds before the simulated C1 of a Total Lunar Eclipse. The pulsing red Eclipse Warning card shows all four contact times. The filmstrip includes a thumbnail captured at the moment of a simulated aircraft transit.

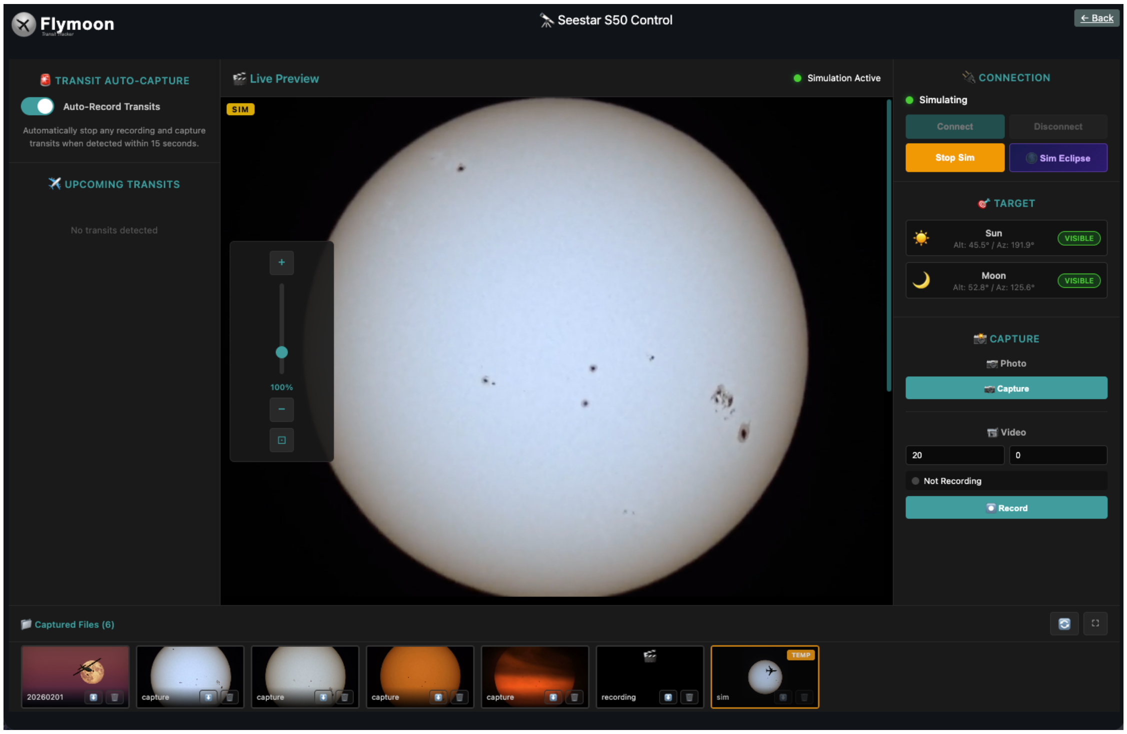
One of Flymoon's most thoughtful design decisions is that the entire eclipse response system can be rehearsed — in compressed time, without waiting for an actual eclipse.

Clicking "Sim Eclipse" injects a set of synthetic contact times into the system: C1 thirty-five seconds in the future, C4 one hundred and five seconds out, with C2 and C3 interpolated for eclipse types that include totality. The user watches in real time as the alert level escalates from Watch to Warning to Active to Cleared, all in under two minutes. The recording system arms, fires, and saves a file to the filmstrip.

The simulation even handles aircraft. During the Active phase, a "Fire Transit" button injects a synthetic aircraft crossing eight seconds away. A plane animation sweeps across the preview. The recording extends. A ✈️ marker appears in the filmstrip.

The simulated recording is real: the browser's `MediaRecorder` API composites the live preview video and the animated aircraft onto a hidden canvas at thirty frames per second, producing a genuine WebM video in which the aircraft is visible crossing the target disk.

The Telescope Control Panel: Ready for the Field



Flymoon simulation mode with the full telescope control interface

Figure 4 — Flymoon in full simulation mode. The telescope panel shows the live (simulated) solar preview stream, complete with visible sunspot groups. The SIM badge confirms simulation mode is active. The connection panel shows the 2×2 button grid — Connect, Disconnect, Simulate, and Sim Eclipse.

Usability has been a constant focus throughout Flymoon's development. The connection panel uses a clean 2×2 grid for its four core actions. Loading spinners stay visible until all flight data, map markers, and table rows have fully rendered.

The telescope connection is robust. A background heartbeat thread sends a keepalive ping to the Seestar every three seconds. If the connection drops, the system attempts reconnection automatically every ten seconds, distinguishing between command timeouts and actual socket failures so that brief response delays don't trigger unnecessary reconnect cycles.

What You Can Capture

A commercial aircraft transits the solar or lunar disk in roughly half a second to two seconds, depending on its altitude, speed, and the angle of crossing. At high magnification the silhouette is surprisingly detailed — you can often resolve the fuselage shape, winglets, and engine nacelles against the bright background. Lunar transits at night, with the Moon in a thin crescent or gibbous phase, reveal the aircraft in striking contrast.

The images that result are genuinely rare. Not because aircraft-solar transits are infrequent — they happen constantly — but because until recently there was no practical way to predict them from an arbitrary ground location. That is exactly the gap Flymoon fills.

Getting Started

Flymoon runs on any machine with Python 3.10 or later, a FlightAware AeroAPI key, and (optionally) a Seestar S50 or S30 on the local network. Setup takes about fifteen minutes:

```
git clone https://github.com/dbetm/flymoon.git
cd flymoon
make setup
```

Edit the generated `.env` file to supply your FlightAware API key and GPS coordinates, then launch with `python app.py`. A browser pointed to `http://localhost:8000` opens the live interface. An interactive configuration wizard (`python3 src/config_wizard.py --setup`) walks through every setting. A pre-built macOS and Windows application bundle is also available for observers who prefer a double-click experience.

Technical Specifications

Component	Technology
Backend	Python 3.10+, Flask
Flight data	FlightAware AeroAPI (ADS-B)
Ephemeris	Skyfield (JPL DE421)
Map	Leaflet.js + OpenStreetMap
Telescope control	Seestar S50/S30 via TCP/JSON-RPC
Frontend	Vanilla JS, CSS3, Web Audio API, MediaRecorder API
Eclipse detection	Skyfield eclipselib + binary-search contact solver
Platform	macOS, Linux, Raspberry Pi

Flymoon is open-source software released under the MIT License. Source code and installation instructions: https://github.com/dbetm/flymoon