Arcade Commander 2.1 — User Guide

Pico-CTR Adalight LED Control Version for the atGames ALU

Version 2.1 | ALPHA
Supported OS: Windows 10/11 & Linux (x86_64)
Components: ArcadeCommanderV2 • ACLighter • ACDispatch

Developed by Mark Abraham

AM4L LLC

Contents

1. What is Arcade Commander?

Arcade Commander is a **specialized lighting control suite built specifically for arcade cabinets**. It replaces static, one-note cabinet lighting with a **fully programmable, high-energy system** capable of reacting dynamically to games, button presses, and automation events. Instead of lights that simply stay on, your cabinet becomes responsive—lighting up with purpose and motion as gameplay unfolds.

The system connects physical arcade controls directly to **dynamic LED effects**, allowing each game to have its **own distinct visual identity**. Buttons don't just illuminate—they communicate gameplay roles, player focus, and system state. This allows lighting to enhance immersion while remaining informative and intentional.

This version is **purpose-built for the atGames Legends Ultimate (ALU) HD cabinet** with Pico-CTR Boards installed and assumes a **2-player control deck**. The application comes **preconfigured for this hardware**, enabling users to begin creating, assigning, and testing lighting effects immediately without the need for manual hardware setup.

Supported Hardware

Current Release:
atGames Legends Ultimate (ALU) HD cabinet with a **2-Player control deck**, using **ACUSTOMARCADE.com's Pico-CTR Controller and LED Controller Boards**.

Future Roadmap:
Planned expansion includes support for **additional controller boards**, **multi-player control decks**, and **fully custom layouts**, allowing Arcade Commander to scale beyond the ALU platform while preserving its structured lighting workflow.

2. System Components

Arcade Commander operates as a **coordinated three-part system**, with each component playing a clearly defined role. Together, these parts work in sync to deliver fast, responsive, and fully automated cabinet lighting—without requiring constant user interaction.

Each component focuses on a specific responsibility, ensuring that lighting design, execution, and automation remain cleanly separated while working seamlessly as a single system.

ArcadeCommanderV2
This is the **primary user interface** and creative control center. It's where configuration, effect design, game assignments, and live testing all take place. From here, users build lighting behaviors, assign them to games, preview results, and send instructions to the lighting engine in real time.

ACLighter
ACLighter is the **lighting engine** responsible for communicating directly with the physical cabinet LEDs. It runs in the background and executes all lighting instructions generated by

ArcadeCommanderV2. Once running, ACLighter continuously listens for commands and applies lighting changes instantly to the real hardware.

ACDispatch

ACDispatch is a **command-line automation helper** designed to integrate with frontends such as LaunchBox or BigBox. Its role is to automatically switch lighting profiles when games start or stop. By passing game information into Arcade Commander, ACDispatch enables seamless, hands-free lighting transitions during normal gameplay.

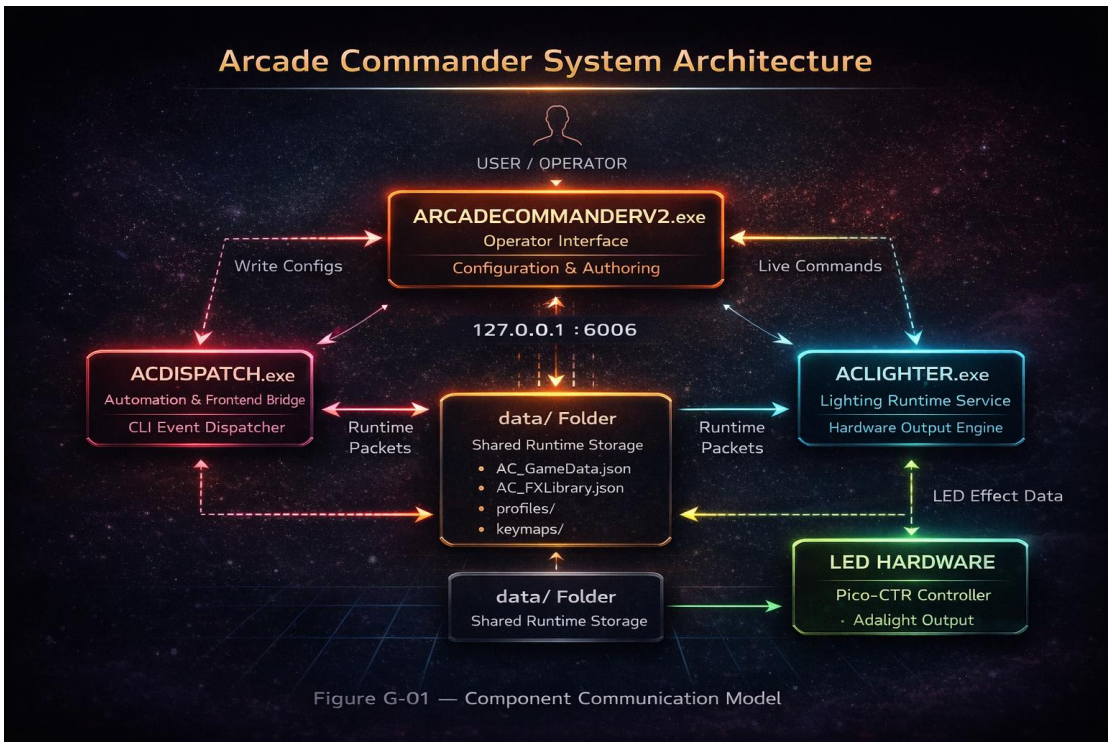


Figure G-01 — Component Communication Model

3. Installation

Getting Arcade Commander up and running is fast and straightforward. There's no traditional installer and no complicated setup — just extract, launch, and go.

You will have to have trust and ignore windows warnings at this point. Getting the application signed comes at the cost of time and money, neither of which I have gotten any money so far to support.

Windows Installation

Download the Windows archive, extract it to a permanent folder (for example, C:\ArcadeCommander\), and run **ArcadeCommanderV2**.

No installer is required.

Once launched, Arcade Commander is immediately ready to communicate with the lighting service and begin controlling your cabinet LEDs.

Linux Installation

Extract the Linux archive to your preferred location.
Ensure all executables have the proper permissions by running:

4. Application Workflow Overview

Arcade Commander is built around a workflow that mirrors how cabinet lighting is created, tested, and used in real gameplay. You configure hardware once, design effects, assign them to games, preview the results, and then let automation take over during normal play.

This structured approach keeps lighting powerful but predictable. Once everything is set up, Arcade Commander handles lighting changes automatically—no manual switching, no constant tweaking.

4.1 Understanding Button Maps, Effects, Animations, and Events

Arcade Commander organizes cabinet lighting into several layered concepts that work together to produce dynamic, game-aware behavior. Separating these concepts keeps lighting reusable, consistent, and fully automated without requiring user interaction during gameplay.

At the foundation is the **Control Deck Layout**, which represents the physical hardware of the arcade cabinet—joysticks, buttons, system controls, and optional devices such as trackballs or spinners. The layout defines *where lighting can exist*.

Built on top of the layout are **Button Maps**, which describe how a specific game uses those physical controls. Different games often use buttons differently—for example, a fighting game versus a platformer. A button map assigns logical meaning to physical buttons so lighting reflects gameplay roles instead of fixed positions.

Once buttons are defined, **Effects** determine *how lighting behaves*. Effects control visual characteristics such as color, brightness, motion, waveform modulation, and sound-driven patterns created in the FX Editor. Effects are reusable lighting behaviors that can be applied across many games and buttons.

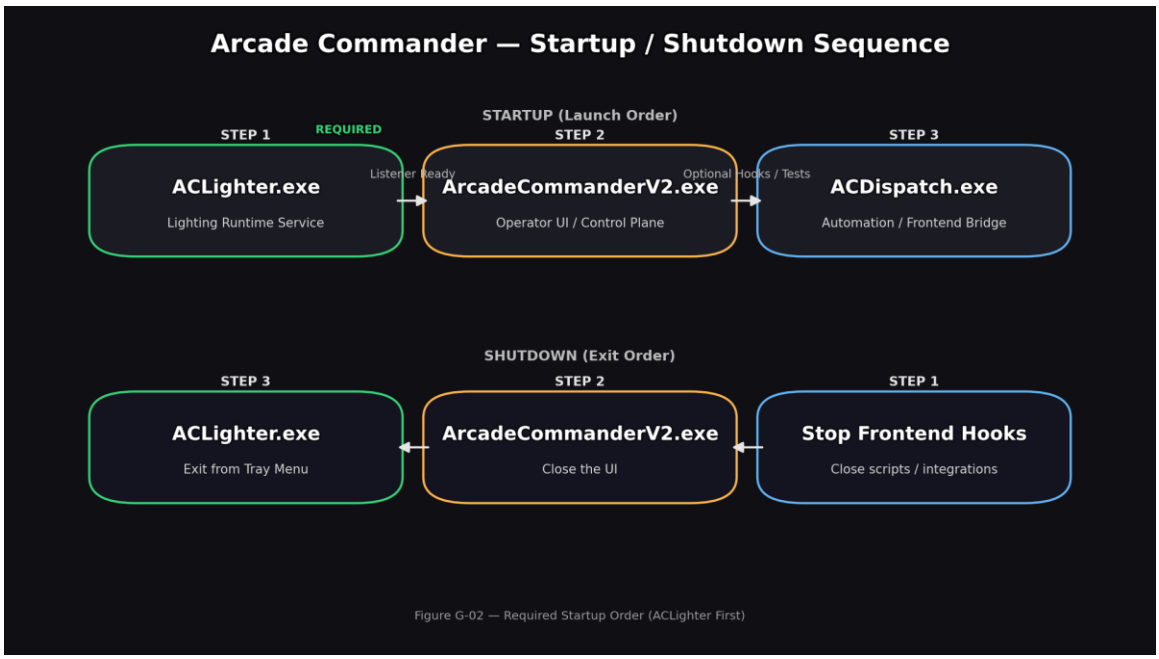
Animations extend effects by combining multiple effects together over time. Instead of a single behavior, animations create sequences—such as an attract pulse followed by a color sweep, or a rhythmic pattern synchronized to audio input. Animations allow lighting to evolve dynamically rather than remain static.

Finally, **Events** determine *when lighting changes occur*. Events are triggers generated by frontend activity or gameplay state, such as when a game starts, exits, pauses, or when the cabinet returns to menu mode. Events automatically activate assigned effects or animations, allowing Arcade Commander to transition lighting without user interaction.

Together, these components form a structured workflow:

- The **Control Deck Layout** defines the physical canvas
- **Button Maps** describe how games use that canvas
- **Effects** define visual behavior
- **Animations** combine behaviors into sequences
- **Events** trigger those sequences automatically

This layered model lets you design lighting once, reuse it across many games, and still maintain consistent, automated behavior.



4.1.5 Understanding Button Maps, Effects, Animations, and Events

Arcade Commander is built around a workflow that mirrors how cabinet lighting is created, tested, and used in real gameplay. You configure hardware once, design effects, assign them to games, preview the results, and then let automation take over during normal play.

This structured approach keeps lighting powerful but predictable. Once everything is set up, Arcade Commander handles lighting changes automatically—no manual switching, no constant tweaking.

4.1.7 Understanding Button Maps, Effects, Animations, and Events

Arcade Commander organizes cabinet lighting into several layered concepts that work together to produce dynamic, game-aware behavior. Separating these concepts keeps

lighting reusable, consistent, and fully automated without requiring user interaction during gameplay.

At the foundation is the **Control Deck Layout**, which represents the physical hardware of the arcade cabinet—joysticks, buttons, system controls, and optional devices such as trackballs or spinners. The layout defines *where lighting can exist*.

Built on top of the layout are **Button Maps**, which describe how a specific game uses those physical controls. Different games often use buttons differently—for example, a fighting game versus a platformer. A button map assigns logical meaning to physical buttons so lighting reflects gameplay roles instead of fixed positions.

Once buttons are defined, **Effects** determine *how lighting behaves*. Effects control visual characteristics such as color, brightness, motion, waveform modulation, and sound-driven patterns created in the FX Editor. Effects are reusable lighting behaviors that can be applied across many games and buttons.

Animations extend effects by combining multiple effects together over time. Instead of a single behavior, animations create sequences—such as an attract pulse followed by a color sweep, or a rhythmic pattern synchronized to audio input. Animations allow lighting to evolve dynamically rather than remain static.

Finally, **Events** determine *when lighting changes occur*. Events are triggers generated by frontend activity or gameplay state, such as when a game starts, exits, pauses, or when the cabinet returns to menu mode. Events automatically activate assigned effects or animations, allowing Arcade Commander to transition lighting without user interaction.

Together, these components form a structured workflow:

- The **Control Deck Layout** defines the physical canvas
- **Button Maps** describe how games use that canvas
- **Effects** define visual behavior
- **Animations** combine behaviors into sequences
- **Events** trigger those sequences automatically

This layered model lets you design lighting once, reuse it across many games, and still maintain consistent, automated behavior.

Relationship Overview

Component	Purpose	Defines	Created In	Used By

Control Deck Layout	Physical cabinet structure	Buttons, joysticks, devices	Controller Config / Designer	Entire system
Button Map	Game control mapping	What each button represents	Game Manager	Effects & Events
Effect (FX)	Lighting behavior	Color, motion, intensity, waveform	FX Editor	Animations & Events
Animation	Timed sequence of effects	Multi-stage lighting patterns	FX Editor	Events
Event	Trigger condition	When lighting activates	Game Manager	Lighting Service

---

Control Deck Relationship

The control deck acts as the physical reference point for all lighting operations. Every lighting action ultimately resolves to a specific LED associated with a real control.

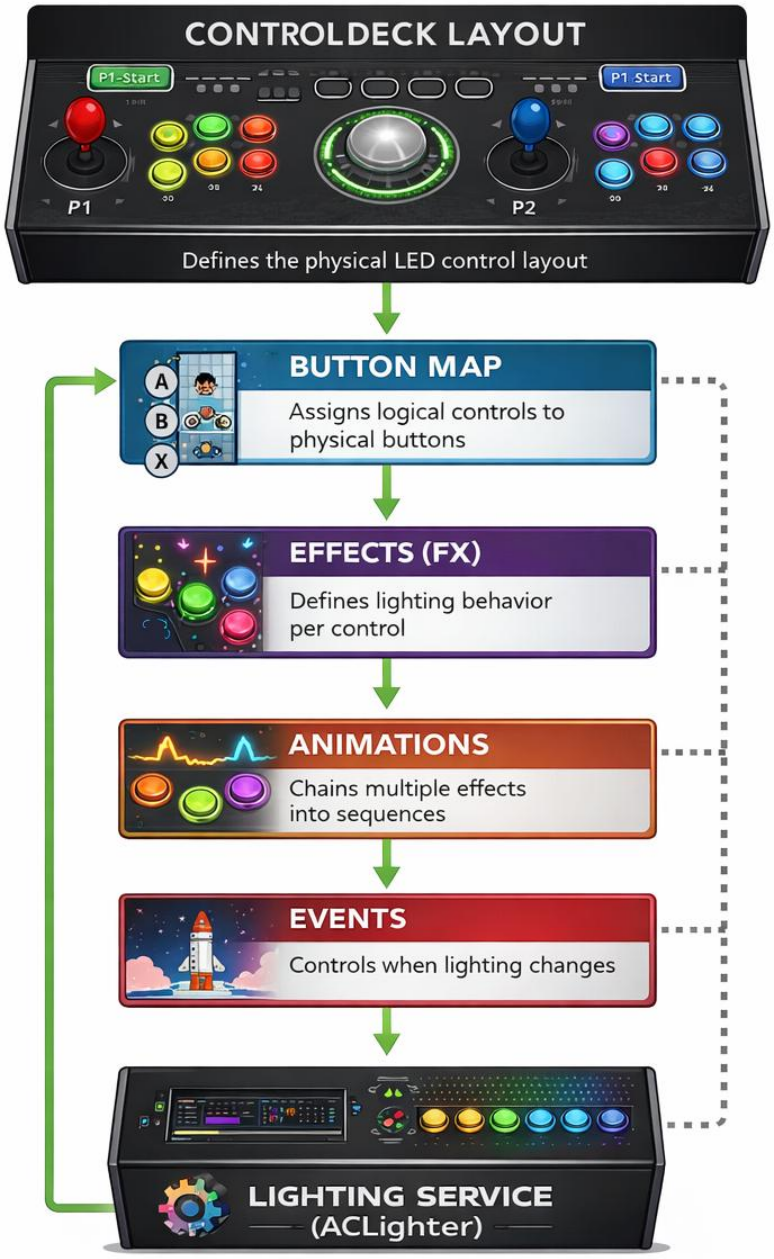
For the ALU cabinet, this typically includes:

- Player 1 buttons and joystick
- Player 2 buttons and joystick
- Start and system buttons
- Admin or menu controls
- Optional devices such as a trackball or spinner (when present)

When a game launches:

- The **GAME_START** event is triggered
- Game Manager loads the assigned Button Map
- The selected Effect or Animation is activated
- The Lighting Service (**ACLighter**) applies lighting to the physical LEDs defined by the Control Deck Layout

This entire process happens automatically and continuously during gameplay.



RELATIONSHIP OVERVIEW

BUTTON MAP	EFFECTS (FX)	ANIMATIONS	EVENTS
Maps Controls	Defines Behaviors	Combines Effects	Triggers Lighting

The control deck acts as the **physical** reference tor all lighting operations. Every lighting action ultimately resolves to a specific **LED** associated with a real control.

Defiles: Cparades, Active. benehvies, play-ring gameptay liske Control Deck...

4.2 Arcade Commander Tab — Live Control

This is the **primary operational screen** shown when Arcade Commander opens. It's where day-to-day interaction happens—selecting games, applying lighting profiles, assigning colors, and testing hardware in real time.

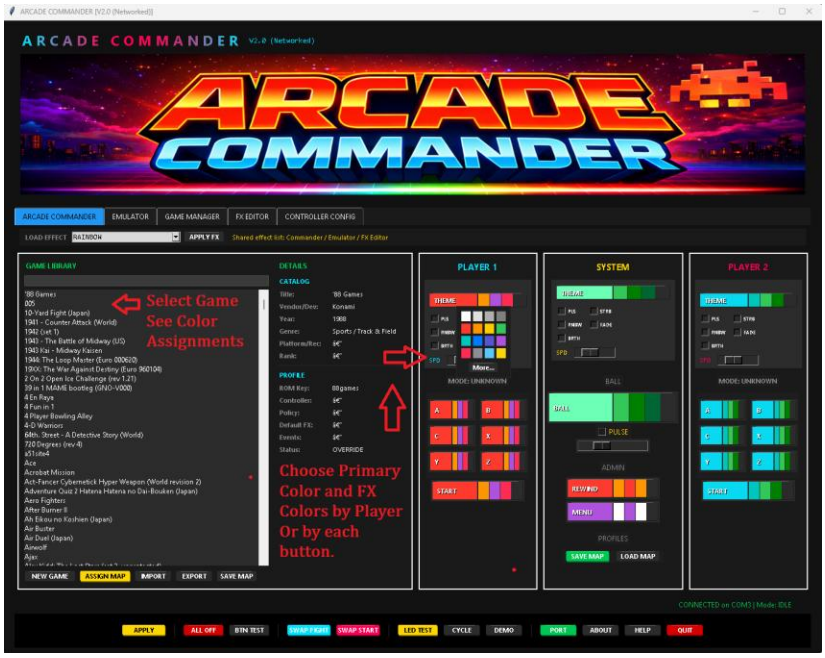
From this screen, changes are sent **immediately** to the lighting service (**ACLighter**) and reflected directly on the physical cabinet. There's no waiting, no reloads, and no guesswork—what you change here is what you see on the machine.

The Arcade Commander tab provides live control over:

- Player colors
- Button assignments
- Global lighting behavior

Built-in LED TEST and BTN TEST tools make it easy to verify that hardware communication is working correctly. These tools allow quick confirmation that LEDs respond as expected and that button mappings are being detected properly before moving on to more advanced configuration.

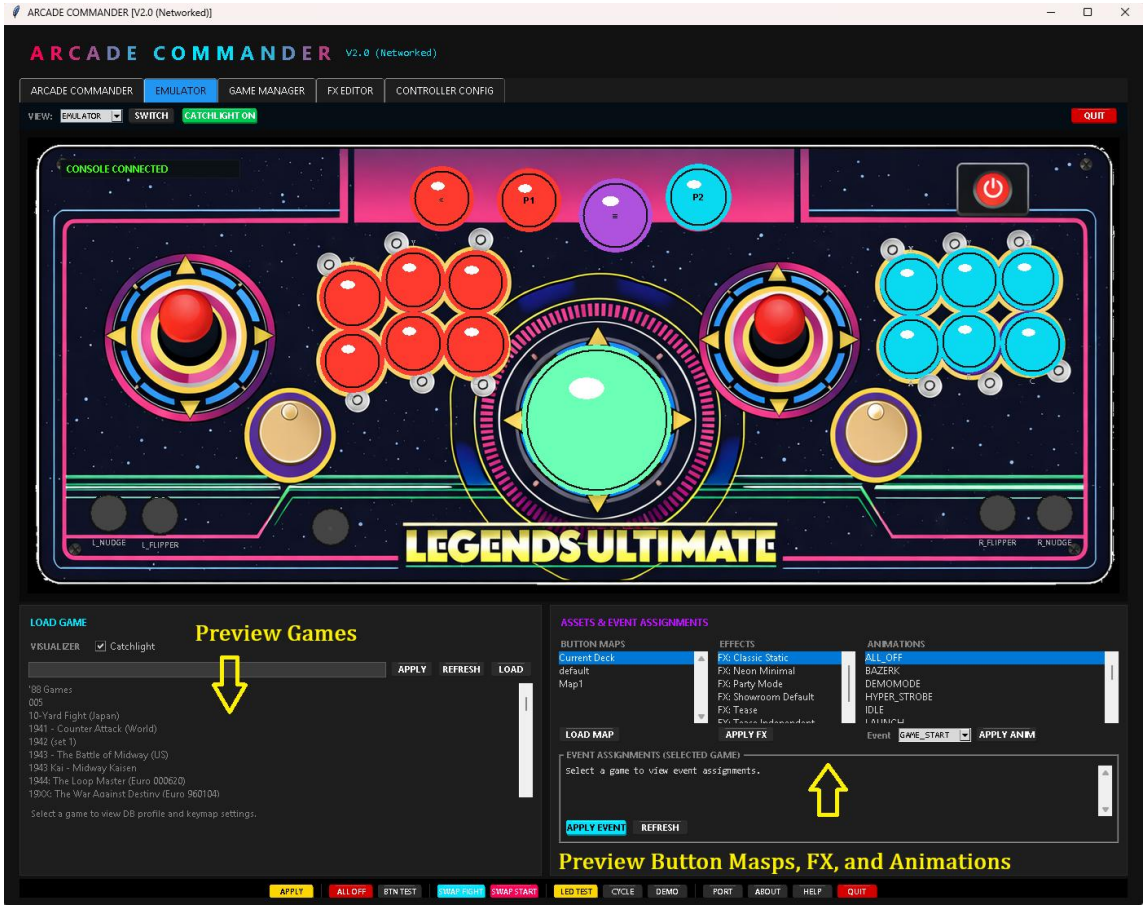
This tab is designed to be fast, responsive, and practical—giving you immediate feedback while configuring or troubleshooting your cabinet lighting.



4.3 Emulator Tab — Preview & Validation

The Emulator provides a virtual representation of the control deck, allowing users to preview button maps, effects, and animations before committing changes. This makes it easy to validate layouts and lighting behavior without immediately relying on physical hardware feedback.

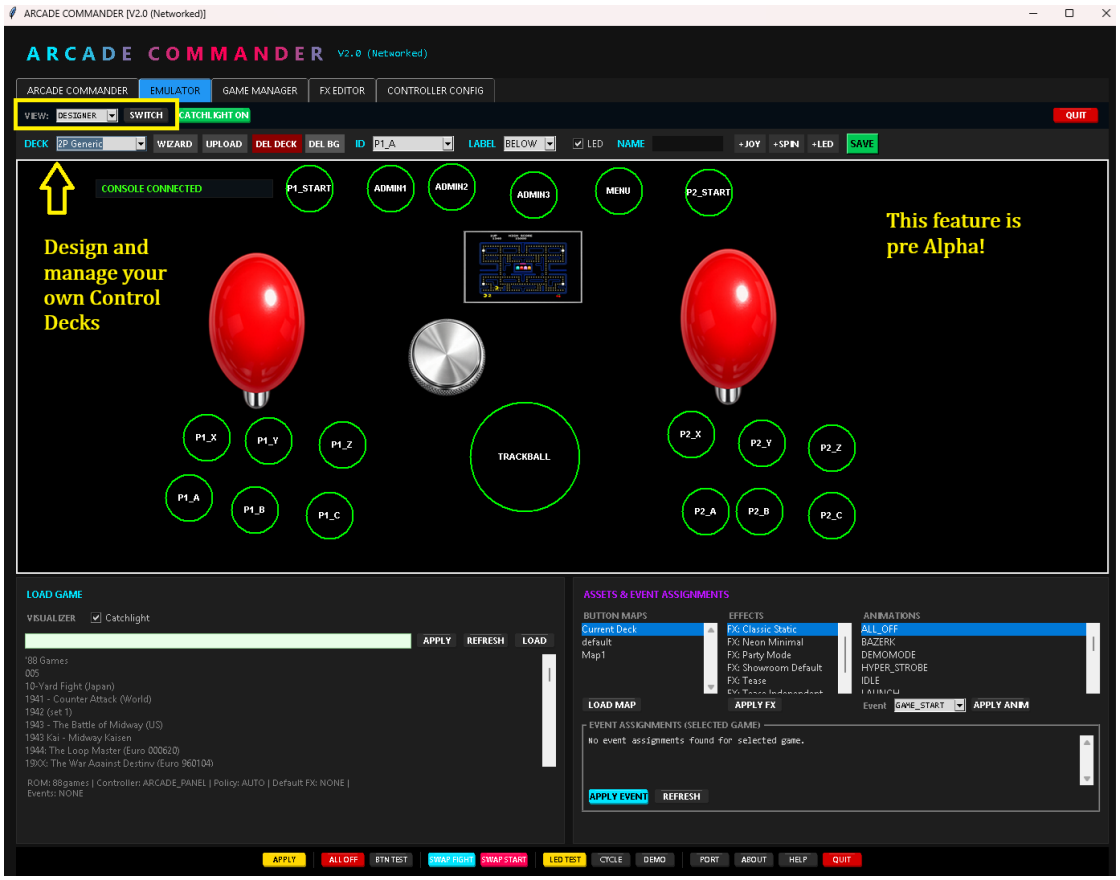
The Emulator can also send lighting output directly to the real control deck through ACLighter, making it both a visualization tool and a live testing environment. This dual role allows users to confidently experiment, adjust, and refine lighting behavior while seeing accurate results.



Designer Mode (Prototype)

Accessible from within the Emulator tab, Designer Mode allows experimental creation of custom control decks. This feature is currently pre-alpha and intended for future expansion beyond ALU hardware.

Designer Mode is provided for exploration and early development purposes and is not yet intended for full production use.



4.4 Game Manager Tab — Game Data & Assignments

The Game Manager is the central database for managing game configurations inside Arcade Commander. This is where games are added, updated, organized, and linked to their specific lighting behavior.

Each game entry can have its own dedicated lighting setup, allowing Arcade Commander to automatically apply the correct button maps, effects, and animations whenever a title is launched.

Lighting assignments can be triggered by a variety of events, including:

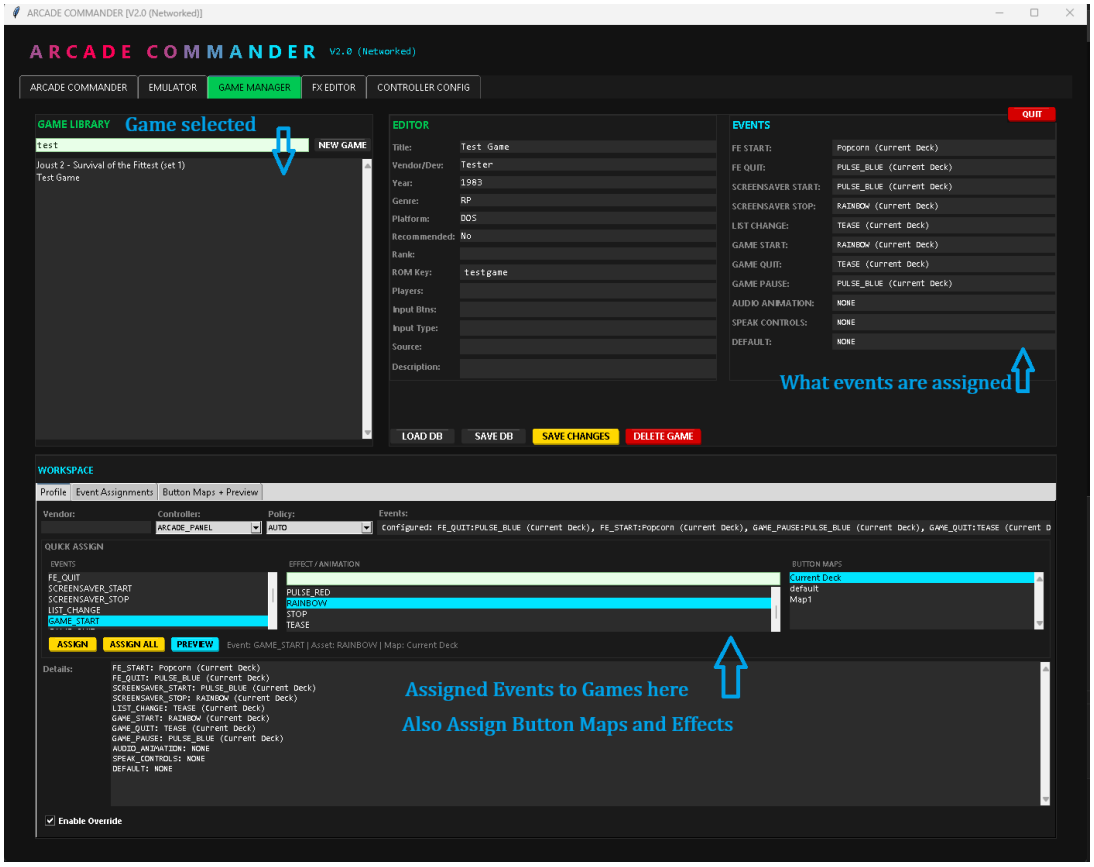
Game start

Game exit

Pause

Screensaver or menu return

Once configured, these assignments activate automatically—no manual switching required during normal gameplay.



Event Assignments

Event assignments define what lighting behavior occurs and when. By tying effects or animations to frontend events, Arcade Commander can seamlessly transition lighting as the cabinet moves between menus, gameplay, and idle states.

This event-driven approach ensures lighting always matches the current system state without requiring user input.

Button Maps + Preview

The Button Maps workspace allows per-button lighting configuration based on how each game uses the control deck. Buttons can be visually assigned and reviewed to ensure the layout matches gameplay expectations.

A built-in preview provides immediate visual feedback, making it easy to confirm button assignments and lighting behavior before launching a game.

4.5 FX Editor Tab — Effect Creation Laboratory

The FX Editor is the creative engine of Arcade Commander. This is where lighting behavior is designed, refined, and brought to life. Users can load sound files, generate waveform-driven effects, adjust intensities and rates, apply multi-color button sets, and chain effects together to create complex lighting animations.

The FX Editor functions like a lighting laboratory, giving you precise control over:

Animation timing

Waveform modulation

Color transitions

Intensity and motion behavior

Effects created here can be reused across multiple games and buttons, making it easy to build a powerful lighting library that stays consistent while still feeling unique from game to game.

By combining effects into animations, users can create lighting sequences that evolve over time instead of remaining static. This allows cabinet lighting to pulse, sweep, react to audio, or transition smoothly between states—adding motion, rhythm, and personality to the control deck.

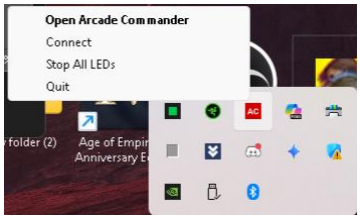
4.6 Controller Configuration Tab — Hardware & Service Settings

The Controller Configuration tab defines cabinet hardware settings and core application behavior. Because Arcade Commander is preconfigured for the ALU cabinet, most users will only need to review the options available here rather than make frequent changes.

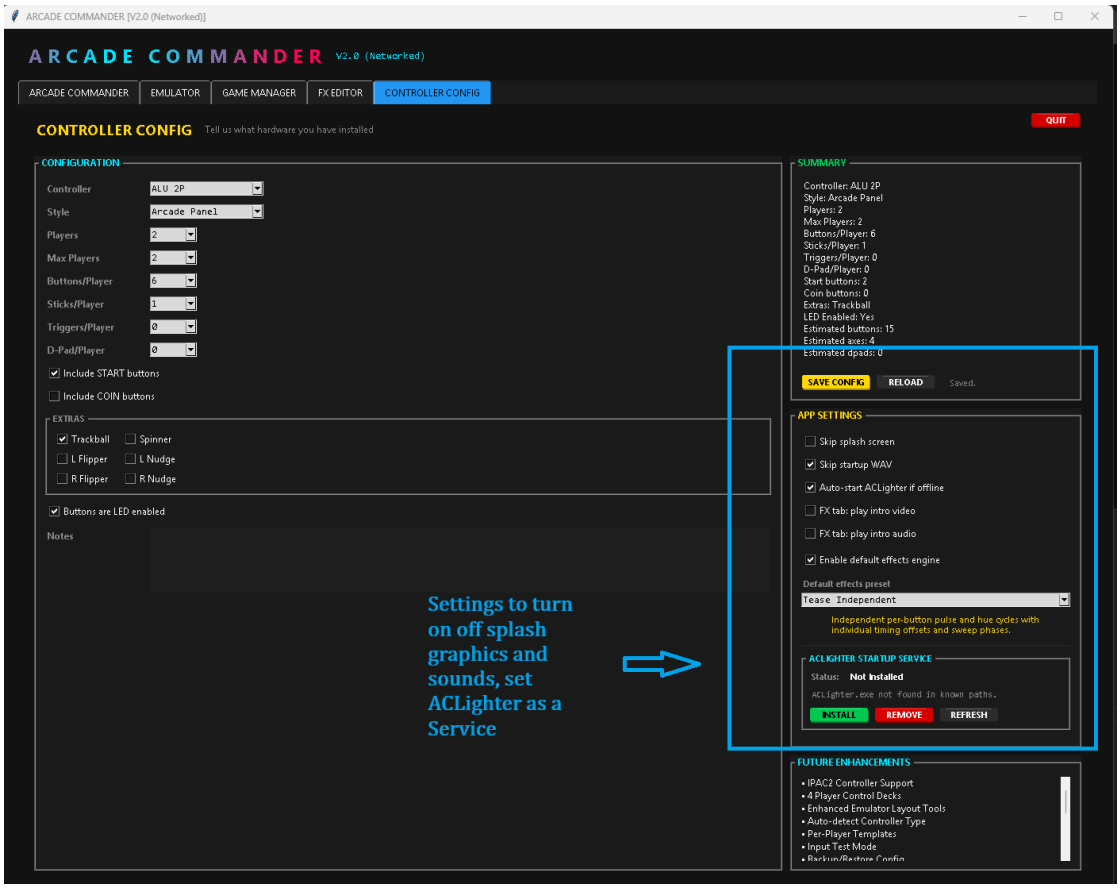
This screen allows you to manage startup behavior, including enabling launch options, controlling splash graphics and sounds, and configuring ACLighter to run automatically as a background startup service.

When enabled, ACLighter runs quietly in the background and can be accessed from the system tray at the bottom of the Windows screen.

Arcade Commander itself can also continue running in the tray when closed using the X, allowing lighting control and services to remain active without keeping the main window open.



At this time, the Configuration screen is intentionally limited. Additional configuration options will be introduced in Version 3, expanding control and customization capabilities as the platform evolves.



5. Automation with ACDispatch

ACDispatch allows frontend software to automatically switch lighting profiles as games start and stop, removing the need for manual lighting changes during normal gameplay.

When configured, ACDispatch works behind the scenes—passing game information into Arcade Commander so the correct lighting setup is activated automatically at the right moment.

Example Configuration

On Game Start:

ACDispatch.exe %romfile%

On Game Exit:

ACDispatch.exe

You can also enable ACDispatch through LED Blinky and trigger events that way if you prefer. The internal database allows you to override default behavior and instead use lighting configurations you have created and assigned to a specific ROM within the Game Manager.

This flexibility allows automation to work exactly the way your cabinet is configured—whether you rely on standard assignments or custom, per-game overrides.

6. Data Storage

All configuration data, profiles, effects, animations, and game assignments are stored inside the data folder. Backing up this directory preserves the entire lighting configuration, making it easy to protect your setup or move it to another system.

Looking ahead to Version 3.0, users will be able to import lighting content created by others, including animations, game data, control deck layouts, and additional shared assets. Arcade Commander has been built with this expansion in mind from the start, ensuring future features integrate cleanly with existing setups.

7. Troubleshooting

If something doesn't look right, don't panic—most issues are quick to diagnose and easy to fix. Use the checks below to get your cabinet lighting back on track fast.

Lights not responding:

Verify that ACLighter is running, then press APPLY to resend lighting instructions.

Effects missing:

Ensure effects were saved correctly to the shared library. Unsaved effects will not appear in assignments.

Automation not working:

Confirm that ROM names match exactly. Automation relies on precise name matching to trigger the correct lighting profiles.

Linux issues:

Verify executable permissions and confirm proper device access rights are set for your system.