Echo Editor User Guide

Version 1.2.0

Welcome to the **Echo Editor**, a graphical user interface (GUI) tool designed to simplify the configuration and setup of your Echo Engine games. Echo Editor allows you to define game metadata (like titles, fonts, and icons), configure narrative elements (such as prolog, cutscenes, and endings), and set core gameplay parameters (health, damage, win conditions) without manually editing text files or code. It handles validation, file copying/conversions, and automatic saving to the appropriate directories in your Echo Engine project structure.

This guide provides a comprehensive walkthrough, including:

- What each field expects as input.
- How to use interactive elements like file pickers, checkboxes, and textboxes.
- Best practices for saving and loading data.
- Troubleshooting common issues.
- Instructions for reporting bugs.

Note: Echo Editor assumes you have an Echo Engine project folder structure set up (e.g., `Working_game/` directory with subfolders like `Text/`, `Fonts/`, `Icons/`, etc.). If you're starting fresh, create these using the Echo Hub.

System Requirements

Hardware

- Minimum: 1024x768 resolution (app starts maximized).
- Recommended: Any modern setup for smooth UI rendering.

Operating System

- Windows

Getting Started

1. **Launch the App**: The App can be launched through the Echo Hub, please ensure a project is open in order to edit.
2. **Icon and Fonts**: If available, the app loads a custom icon and attempts to use "NovaFont" (falls back to Arial).
3. **Navigation**: Use the tab view at the top to switch between sections:
    - **Game Setup**: Core game configuration (main focus of this guide).
    - **Tutorial**: Edit tutorial levels (grid-based editor).
    - **Main Level**: Configure main gameplay areas.
    - **Export**: Package your game for distribution.
    - **Return to Hub**: Navigate back to a central hub (if integrated).
    - **Test App**: Run a preview of your configured game.
    - **About**: App info and credits.
    - **Help**: Built-in help (this document).
4. **Tooltips**: Hover over labels for contextual help (e.g., "Initial player health - Defaults to 1").
5. **Saving**: Click "Save" at the bottom of each tab. Validation runs automatically—fields turn red for errors.
6. **Loading**: On startup (and after save), existing files are auto-loaded and highlighted in dark green (#006400) to indicate "loaded" status.

Pro Tip: Always save after changes. The app creates directories as needed (e.g., `../Working_game/Text/Stories/`) but won't overwrite unrelated files.

Detailed Guide: Game Setup Tab

This tab is the heart of the editor. It's divided into scrollable sections: **Basic**, **Text**, and **General Gameplay**. Use the scrollbar on the right to navigate long forms.

Input Types Explained

- **Text Entries**: Free-form or numeric fields. Validation happens on focus-out (tab away or click elsewhere).
- **File Pickers**: Click the 📂 button to browse files. Supported extensions are validated (e.g., .ttf for fonts).
- **Text-or-File Widgets** (for narrative sections):
    - Checkbox for "File Path" (load from .txt) or "Text" (type directly).
    - Mutual exclusive: Selecting one unchecks the other.
    - Validation: Ensures .txt for files; non-empty for text.
- **Error Indicators**: Red text below fields (e.g., "File not found"). Fields turn red (#661111) on error, default gray (#444444) when valid.
- **Integers Only**: Fields like health auto-validate for positive digits.

Basic Section

These set foundational game assets. Files are copied/converted to `../Working_game/` subdirs.

Field	What to Enter	How to Use	Validation/Tips	Save Location
Title	Game name (e.g., "Echo Quest").	Type directly.	Required; no special chars needed.	`../Working game/Text/Misc/Title.txt`
Font	Path to main game font.	Click 📂 to pick .ttf. Optional (falls back to default).	.ttf only; file must exist. Copied as-is.	`../Working game/Fonts/Font.ttf`
Icon	Path to game icon/image.	Click 📂 to pick .png/.jpg/.jpeg/.ico. Optional.	Converts to PNG; must be valid image.	`../Working game/Icons/Icon.png`
Music	Path to background track.	Click 📂 to pick .mp3/.wav/.ogg. Optional.	Converts MP3/OGG to WAV (requires FFmpeg if installed). .WAV copies directly.	`../Working game/Sounds/Background.wav`
Credits	Credits to display	Checkbox: File (📂 for .txt) or Text (type in box).	Required; .txt or non-empty text. Encoding: UTF-8.	`../Working\ game/Text/Misc/Credits.txt`

Usage Notes:

- Fonts: Use TrueType (.ttf) for compatibility. Test in-game for rendering issues.
- Icon: Square images (e.g., 256x256) work best. The app resizes/previews if needed.
- Music: Loopable tracks recommended. If conversion fails, provide .wav to avoid errors.

Text Section

Narrative content for story beats. Each uses a **text-or-file widget**:

- **File Path Mode**: Pick a .txt (content copied verbatim).
- **Text Mode**: Type multi-line text (saved directly).

Field	What to Enter	How to Use	Validation/Tips	Save Location
Prolog Text	Intro story (e.g., "In a distant echo...").	Checkbox: File (📂 for .txt) or Text (type in box).	Required; .txt or non-empty text. Encoding: UTF-8.	`../Working game/Text/Stories/Prolog/Prolog.txt`

Field	What to Enter	How to Use	Validation/Tips	Save Location
Cutscene Text	Transition from tutorial to main game.	Same as above.	Required; use line breaks for dialogue.	`../Working game/Text/Stories/Tutorial/Tutorial_completed.txt`
Game Over Text	Loss screen message.	Same as above.	Required; keep motivational (e.g., "Try again!").	`../Working game/Text/Stories/Ending/Game_over.txt`
Win Text	Victory epilogue.	Same as above.	Required; spoiler-free if sharing.	`../Working game/Text/Stories/Ending/Win.txt`

Usage Notes:

- **Switching Modes**: Check "File Path" or uncheck to enable Text (auto-toggles).
- **Multi-Line**: Textbox supports Enter for new lines; preserved on save.
- **External Editing**: Write in a text editor, then load via file picker.
- **Loading Existing**: If file exists and has content, auto-loads into Text mode (green highlight).

General Gameplay Section

Core mechanics. Integers and comma-separated lists.

Field	What to Enter	How to Use	Validation/Tips	Save Location
Base Health	Starting HP (e.g., "3").	Type integer. Defaults to 1 if empty.	Required; positive integer (>0).	`../Working game/Finishing/Default_health.txt`
Damage Chance	Denominator for 1/x risk (e.g., "2" for 50%).	Type integer. No damage in tutorial.	Required; positive integer (>0).	`../Working game/Finishing/Damage_chance.txt`
Win Location	Room coordinates (e.g., "5,3,-2").	Comma-separated X,Y,Z (Does not support negatives).	Required; exactly 3 integers, no extra commas (e.g., no ",," or leading/trailing). Saved one per line.	`../Working game/Finishing/Required_room.txt`
Win Items	Required collectibles (e.g., "Key,Sword,Key2").	Comma-separated names.	Optional; trims spaces, ignores empties. Saved one per line.	`../Working game/Finishing/Required_items.txt`
Tutorial Items	Items to complete tutorial (e.g., "Map,Compass").	Comma-separated names.	Optional; same as above.	`../Working game/Tutorial/Required_items.txt`

Usage Notes:

- **Damage Chance**: Fraction is 1/[your value]. E.g., 4 = 25% per room entry.
- **Coordinates**: X,Y,Z for 3D space. Validate format to avoid crashes.
- **Items**: Case-sensitive; match your game's item IDs. No spaces in names (use underscores if needed).

Saving in Game Setup

- **Validate & Save Button**: Bottom-center. Runs full validation:
  - Required fields checked (e.g., Title, Health).
  - File existence/types enforced.
  - Formats parsed (e.g., coords split/joined).
- **Success**: Green dialog; fields turn green if loaded.
- **Errors**: Red dialog lists issues (e.g., "Win Location: Must have exactly three numbers").
- **Auto-Load on Start**: Pre-fills from existing files.

Advanced Tip: Empty optional fields delete the target file (e.g., no music = remove Background.wav).

Other Tabs: Overview

Tutorial Tab

A grid-based editor for tutorial levels (single-floor, 40x40 max grid).

- **Grid Canvas**: Click cells to place rooms/items (background #333333 hides lines for clean borders).
- **Info Display**: Right panel shows selected cell details (e.g., room type).
- **Plus Buttons**: Add rows/columns dynamically.
- **Save**: Bottom button saves grid state to tutorial files.
- **Usage**:
  - Hover for tooltips.
  - Clear info via internal functions (e.g., for selected cells).
  - Focus on placing tutorial items from Game Setup.

Tip: Grid state is stored in memory. Save for persistence.

Main Level Tab

Similar to Tutorial but for core levels. Configure multi-floor grids, damage text, etc.

- **Save**: Validates and saves level data.

- **Usage**: Click to build rooms and layout. Link to Win Location from Game Setup.

Export Tab

- Packages your `Working_game/` into a distributable (e.g., ZIP).
- **Fields**: Output path, include assets checkbox.
- **Usage**: Click "Export" after full setup. Handles dependencies like fonts/music.

Return to Hub Tab

- Integrates with Echo Engine's hub system.
- **Usage**: Button to launch hub and close the editor.

Test App Tab

- Previews your game with current config.
- **Usage**: Click "Run Test" to launch a preview (subprocess call).
- **Tip:** Fixes in setup first—crashes here indicate validation misses.

About Tab

- Credits, version (v1.1.0), links to Nova Foundry.
- **Usage**: Read-only; click links to open browser.

Help Tab

- Embeds this document or links to GitHub wiki.
- **Usage**: Searchable text for quick ref.

Functions and Advanced Usage

Core Features

- **File Pickers (📂)**: Integrated browser for assets; auto-validates extensions.
- **Text-or-File Widgets**: Toggle between loading .txt or typing; mutual exclusive checkboxes.
- **Validation System**: Real-time on blur; full check on save. Handles copies, conversions (e.g., MP3 to WAV), and parsing (e.g., commas to newlines).
- **Auto-Highlighting**: Green for loaded content; resets on empty.
- **Tooltips**: Hover for inline help on all labels/fields.

- **Image Preview**: Scaled display for icons (via internal canvas).

User Tip: For large texts, use external .txt files to avoid textbox limits. All saves use UTF-8.

Saving, Loading, and Best Practices

- **Auto-Save on Valid**: No—manual "Save" only, but loads on open.
- **Backup**: Copy `Working_game/` before experiments.
- **Batch Edits**: Load .txt externally, then import.
- **Validation Order**: Fix reds top-to-bottom; coords/items are picky.
- **Performance**: For 100+ items, split lists or use files.

Troubleshooting

Issue	Cause	Fix
Red "File not found"	Path invalid.	Re-pick via 📂; check exists.
Music conversion fails	No FFmpeg.	Provide .wav; install FFmpeg separately if needed.
Font not loading	Wrong .ttf.	Test .ttf in OS; fallback to Arial.
Coords error	Extra commas.	E.g., "1,2,3" only—no "1,,2".
Empty green fields	Loaded empty .txt.	Delete file or re-save.
UI freezes	Large image/text.	Resize icons <1MB; split texts.
Icon not set	Path wrong.	Use absolute path or fix location.

Logs: Check any console output if launched from terminal.

Reporting Bugs

Encounter a crash, validation bug, or missing feature? Help improve Echo Editor by reporting issues on GitHub:

- **Repository**: https://github.com/DirectedHunt42/EchoEngine/issues
- **What to Include**:
    - Steps to reproduce (e.g., "Enter '1,,2' in Win Location").
    - OS version.
    - Screenshot/error dialog.

Your feedback drives updates—thanks for building with Echo!

Last Updated: November, 2025.