Inspector Designer Pro

Visual Inspector Customization for Unity

User Guide — v1.1

Design beautiful, functional inspectors without writing a single line of editor code.

Requires Unity 2021.3 LTS or later | Editor-Only | Zero Build Impact

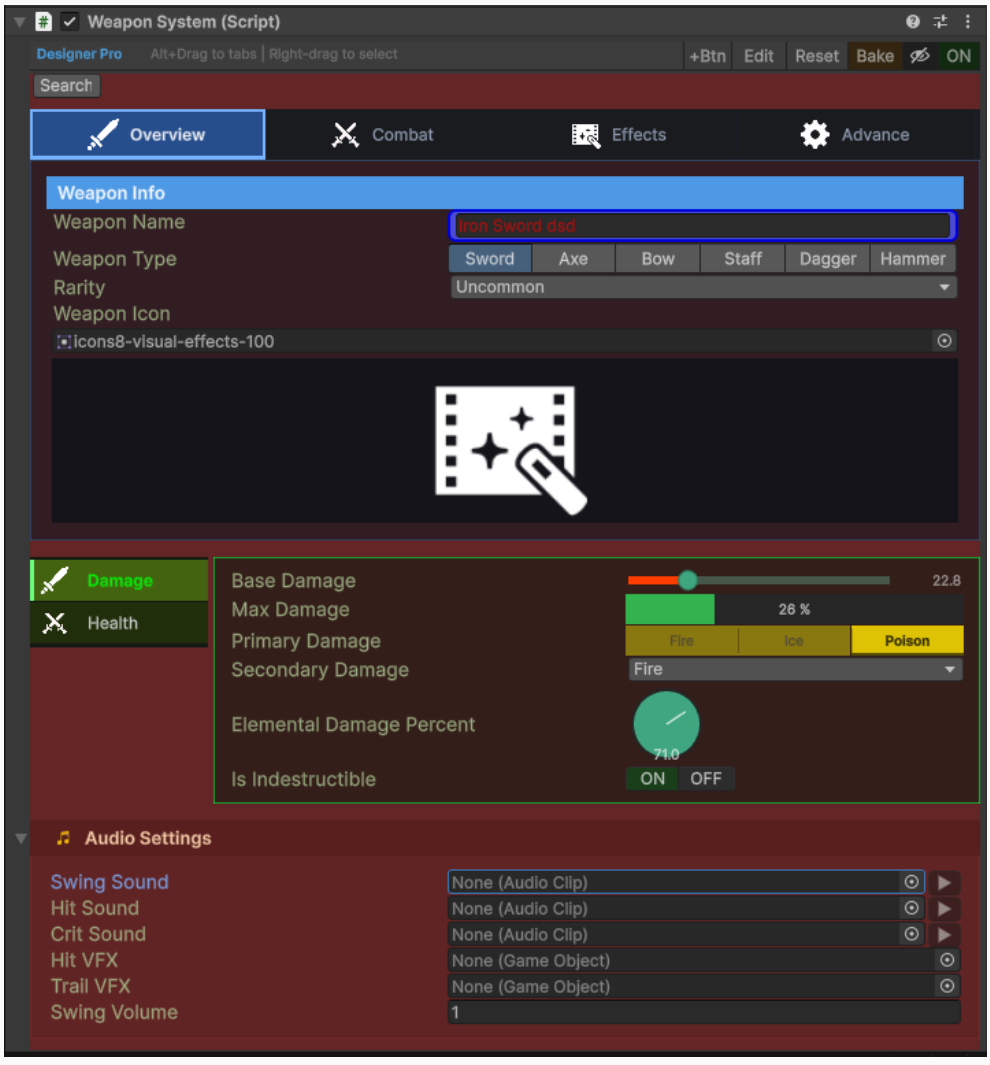


Fig 0.1 — Overview: A custom inspector designed with Inspector Designer Pro

Table of Contents

1. What is Inspector Designer Pro?
2. Installation
3. Quick Start — Your First Custom Inspector
4. Mini Property Popup
5. Full Editor Popup
6. Designer Window
7. Field Type Overrides
8. Tab Groups
9. Foldout Groups
10. Custom Buttons
11. Marquee Multi-Select & Bulk Actions
12. Search & Pinned Fields
13. Preset Themes
14. Undo Support
15. Profile System
16. Technical Details
17. Troubleshooting & FAQ
18. Support & Contact

1. What is Inspector Designer Pro?

Inspector Designer Pro is a Unity Editor extension that gives you complete visual control over how your MonoBehaviour inspectors look and behave — without writing any C# editor code.

Unity's default inspector shows all serialized fields in a flat, unstyled list. For complex components with dozens of fields, this becomes difficult to navigate and unprofessional to present. The traditional solution is writing custom Editor classes and PropertyDrawers — time-consuming work that adds maintenance burden and potential build issues.

Inspector Designer Pro eliminates this entirely. Through right-click menus, popup windows, and a dedicated Designer Window, you can:

- Restyle any field with custom colors, fonts, icons, bold, italic, and alignment
- Override how fields display — turn a float into a slider, knob, or progress bar
- Organize fields into tabbed layouts and collapsible foldout groups
- Add visual structure with headers, separators, spaces, and box groups
- Add clickable buttons that call methods on your component
- Set up conditional visibility so fields show/hide based on other values
- Add validation rules with visual error feedback
- Apply preset themes with one click
- Preview AudioClip fields without entering Play mode

All customization is stored in lightweight ScriptableObject profiles — one per MonoBehaviour type.

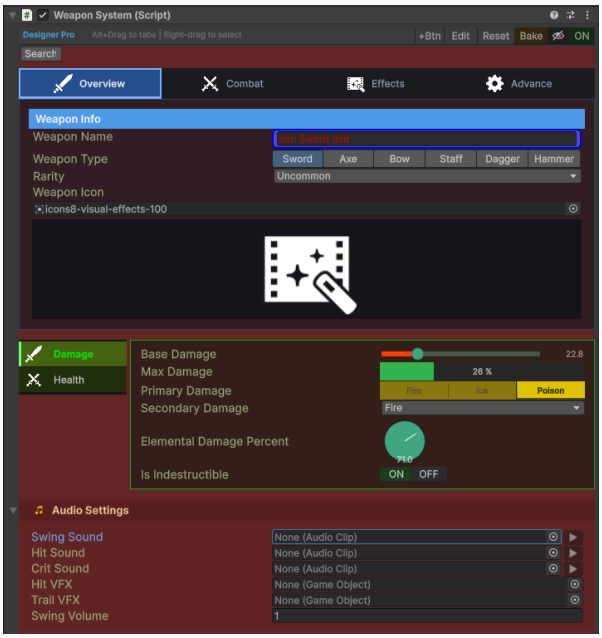


Fig 1.1 — A complex component inspector customized with tabs, foldouts, styled fields, and custom buttons

2. Installation

From the Unity Asset Store

1. Open Unity and go to Window > Package Manager.
2. Click My Assets and find Inspector Designer Pro.
3. Click Download, then Import.
4. Accept all files in the import dialog and click Import.

Verifying Installation

After importing, verify the installation:

- Go to Window menu — you should see Inspector Designer Pro listed.
- Select any GameObject with a MonoBehaviour — right-click any field in the inspector. You should see the Inspector Designer Pro context menu.
- Check that the folder Assets/Inspector Designer Pro/Editor exists.

[Scree

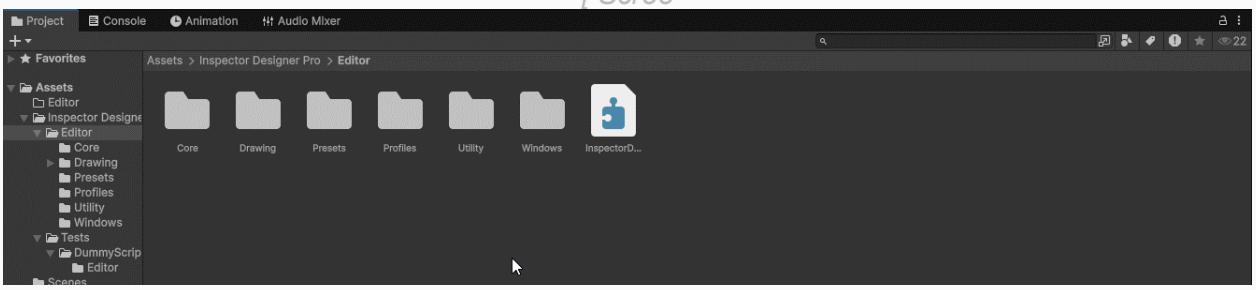


Fig 2.1 — The Window menu showing Inspector Designer Pro entry

> **Note:** Inspector Designer Pro is editor-only. It will not appear in builds or affect runtime performance in any way.

3. Quick Start — Your First Custom Inspector

Let's customize a component inspector in under 60 seconds.

Step 1: Select a GameObject

Select any GameObject in your scene that has a MonoBehaviour component with serialized fields.

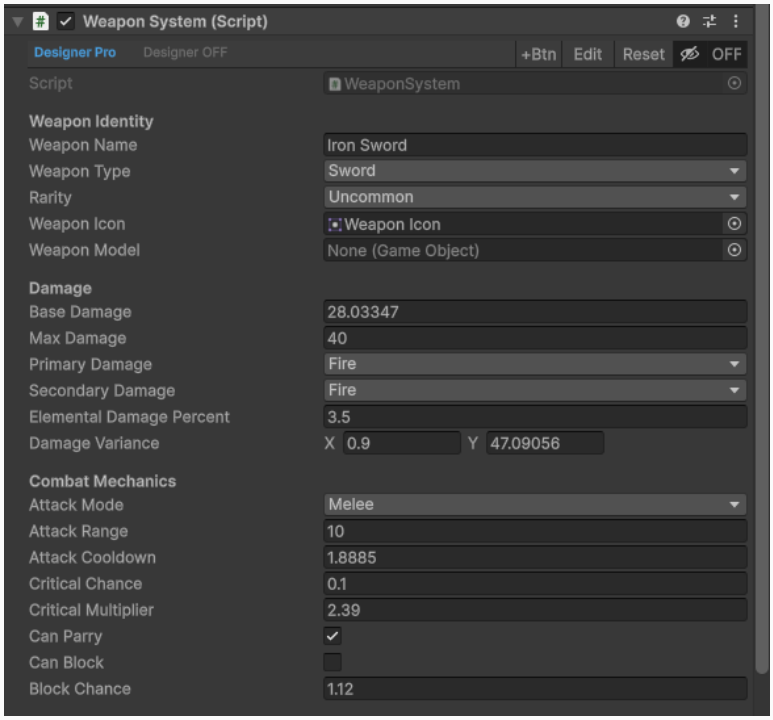


Fig 3.1 — A default Unity inspector showing a MonoBehaviour with several fields

Step 2: Right-Click a Field

Right-click on any field label in the inspector. The Inspector Designer Pro context menu appears with all available options.

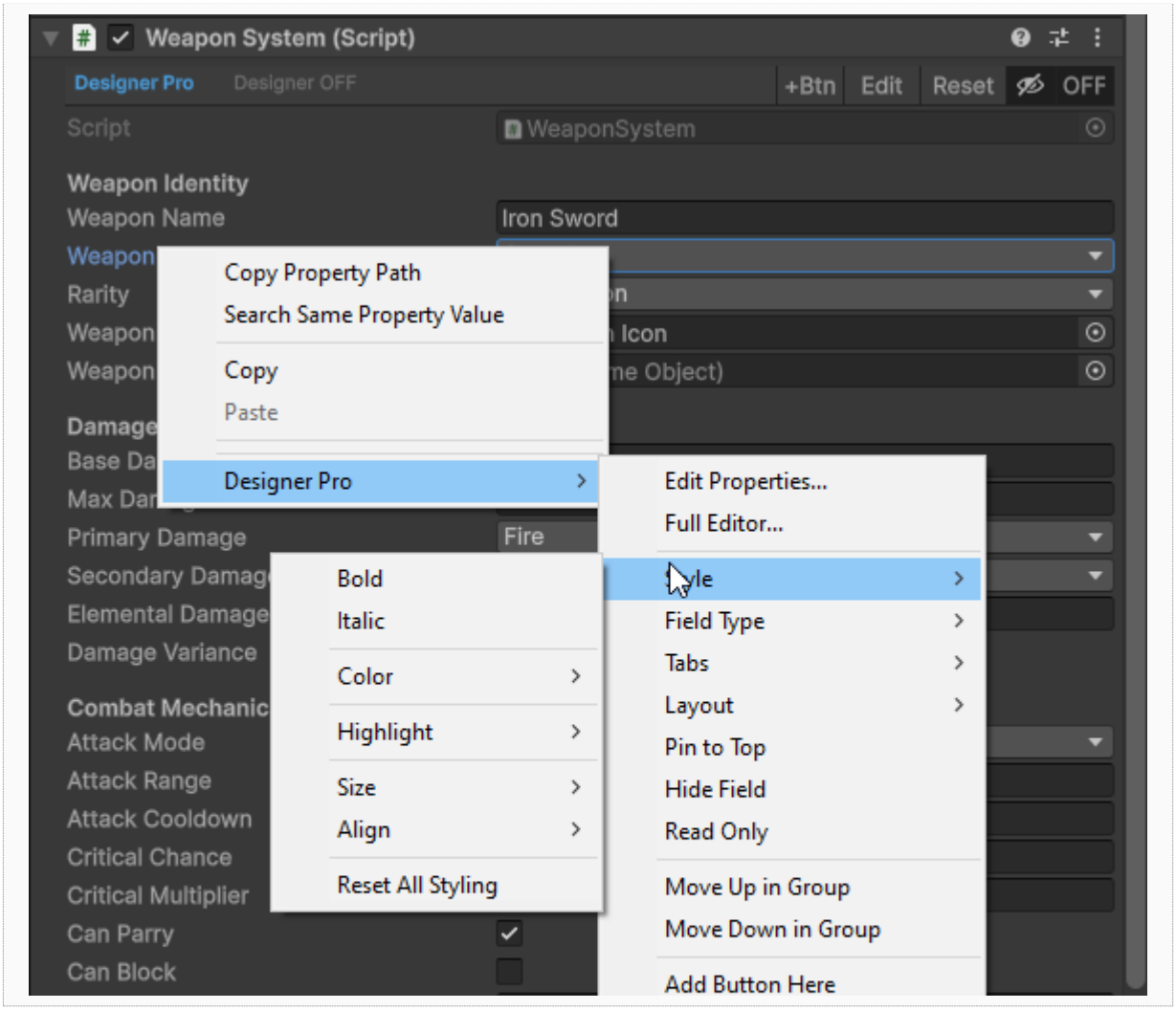


Fig 3.2 — The right-click context menu showing all available options

Step 3: Make Changes

Try any of these quick actions:

- Click Style to give the field a custom Styling like Bold, Italic,Font Color, Font Style ect.
- Click Open Mini Popup for a compact settings panel.

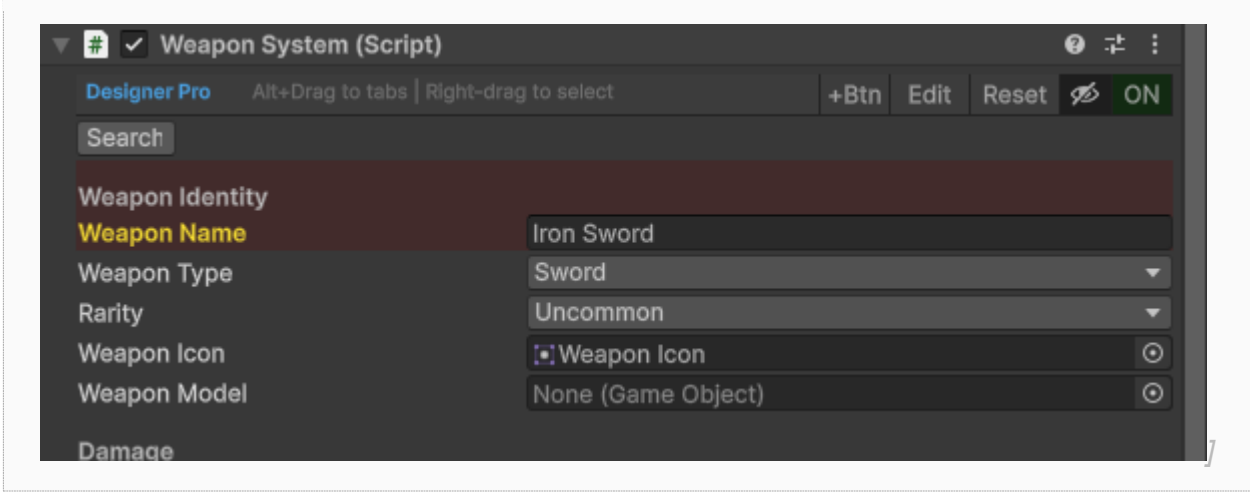


Fig 3.3 — The inspector after changing font color, size, style, and highlighting

Step 4: Changes Save Automatically

Every change is automatically saved to a ScriptableObject profile asset in your project. The profile is created the first time you customize any field on that MonoBehaviour type. You can find profiles in your project using the Search bar — they are named after the target component type.

> **Tip:** All changes support full Undo (Ctrl+Z). Don't worry about experimenting — you can always undo.

4. Mini Property Popup

The Mini Property Popup is a compact, single-click settings panel. Right-click (then choose open Designer Pro -> Edit Properties) on any field label to open it.

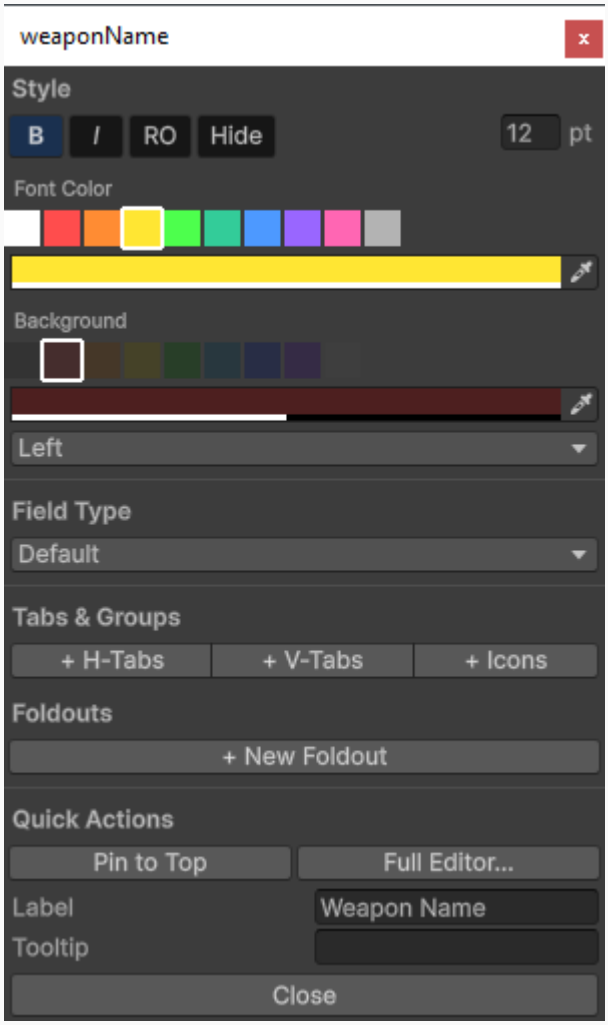


Fig 5.1 — The Mini Property Popup showing field settings

What You Can Configure

The Mini Popup shows the most commonly used settings in a compact layout:

- **Styling—** Custom styling for the field
- **Field Type —** Override how the field renders (with smart type filtering)
- **Tab Group Membership —** Toggle checkboxes to add/remove the field from any tab group. Can also create a new tab group.
- **Foldout Group Membership —** Toggle checkboxes to add/remove the field from any foldout group. Can used to create a new foldout group.

- **Pin To Top :-** Will Pin the property at the top of inspector**.**

Foldout Styling from Mini Popup

Each foldout group row has a small "S" (Style) button. Click it to expand inline styling options for that foldout:

- Header background and text color
- Font size and bold toggle
- Icon picker — click to browse and select from Unity's built-in icon library
- Content area background color
- Content border color, border width, and padding

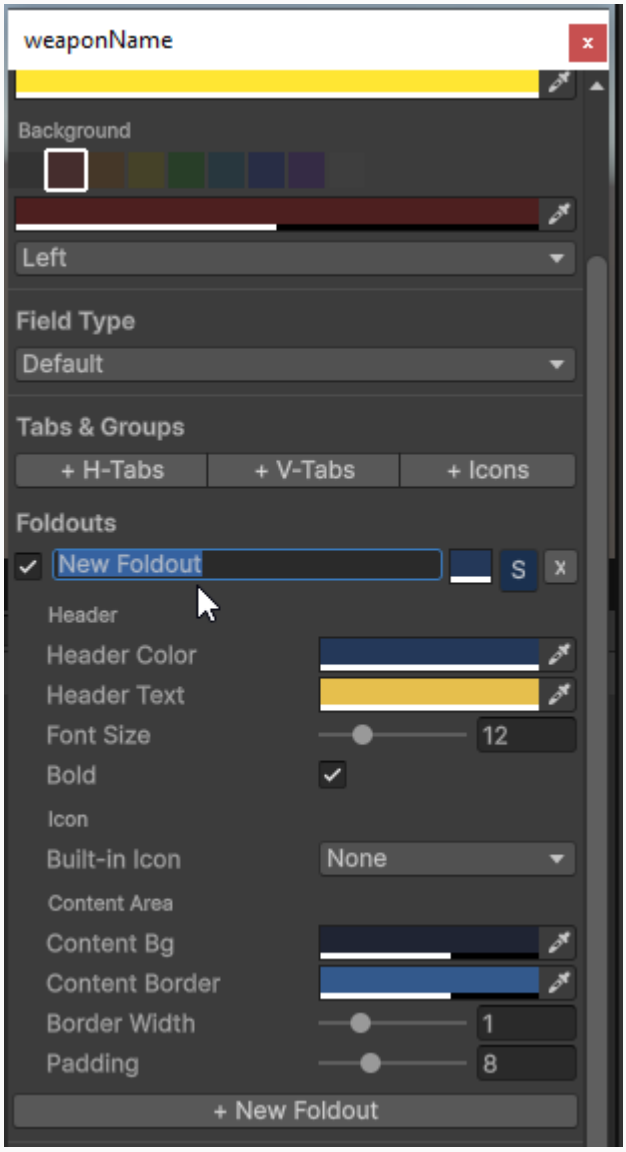


Fig 5.2 — Foldout styling expanded within the Mini Popup, showing icon picker and color fields

Opening the Full Editor

At the bottom of the Mini Popup, click Full Editor... to open the complete Field Customization Popup with all available options.

5. Full Editor Popup

The Full Editor Popup (Field Customization Popup) gives you complete control over every aspect of a field. Access it via right-click >Designer Pro > Full Editor... or the button at the bottom of the Mini Popup.

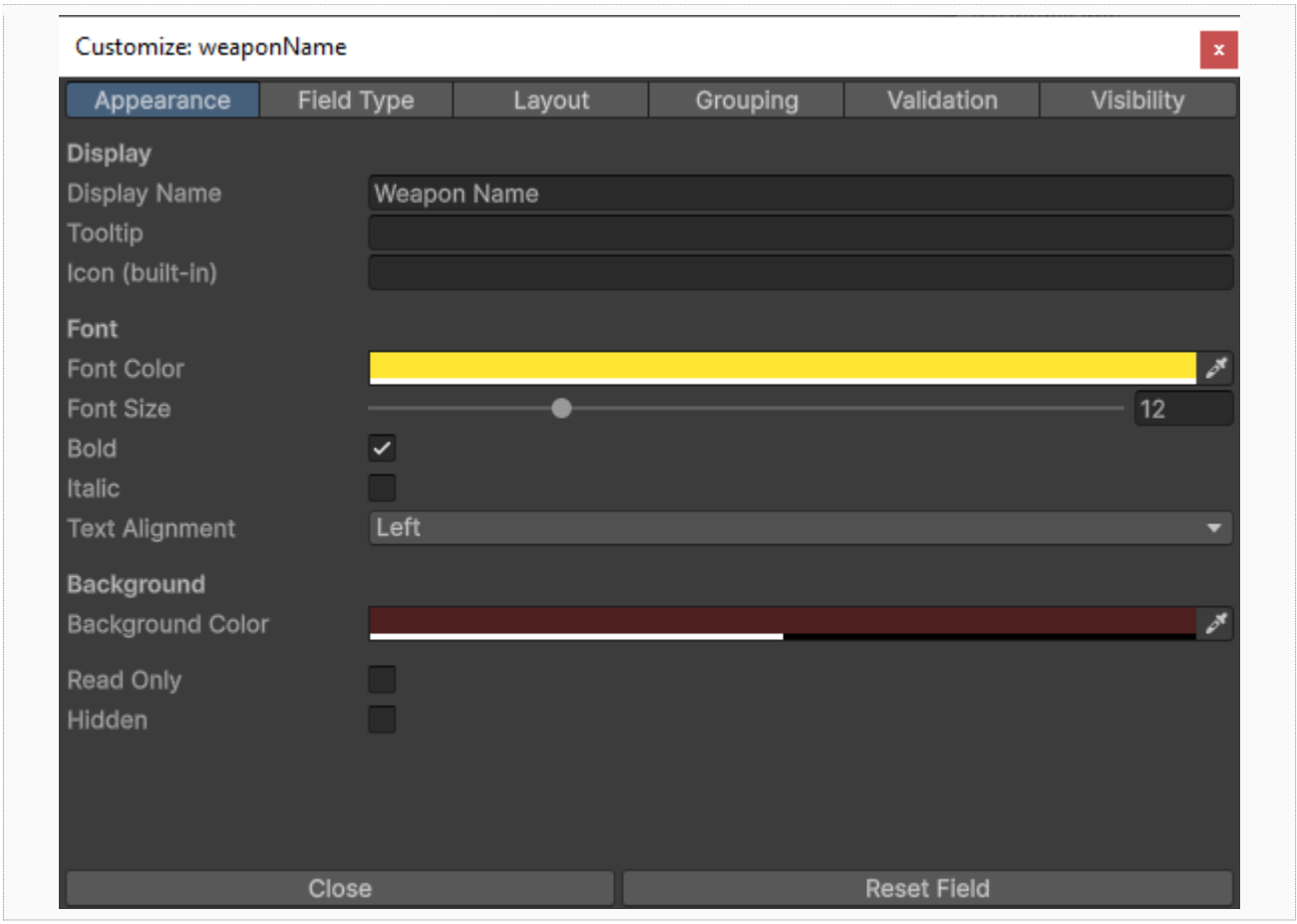


Fig 6.1 — The Full Editor Popup with all tabs visible

Tabs in the Full Editor

The popup is organized into tabs:

Appearance Tab

Controls how the field looks:

- Display name and tooltip
- Font color, font size, bold, and italic toggles
- Background color
- Text alignment (Left, Center, Right)
- Icon selection from Unity built-in icons

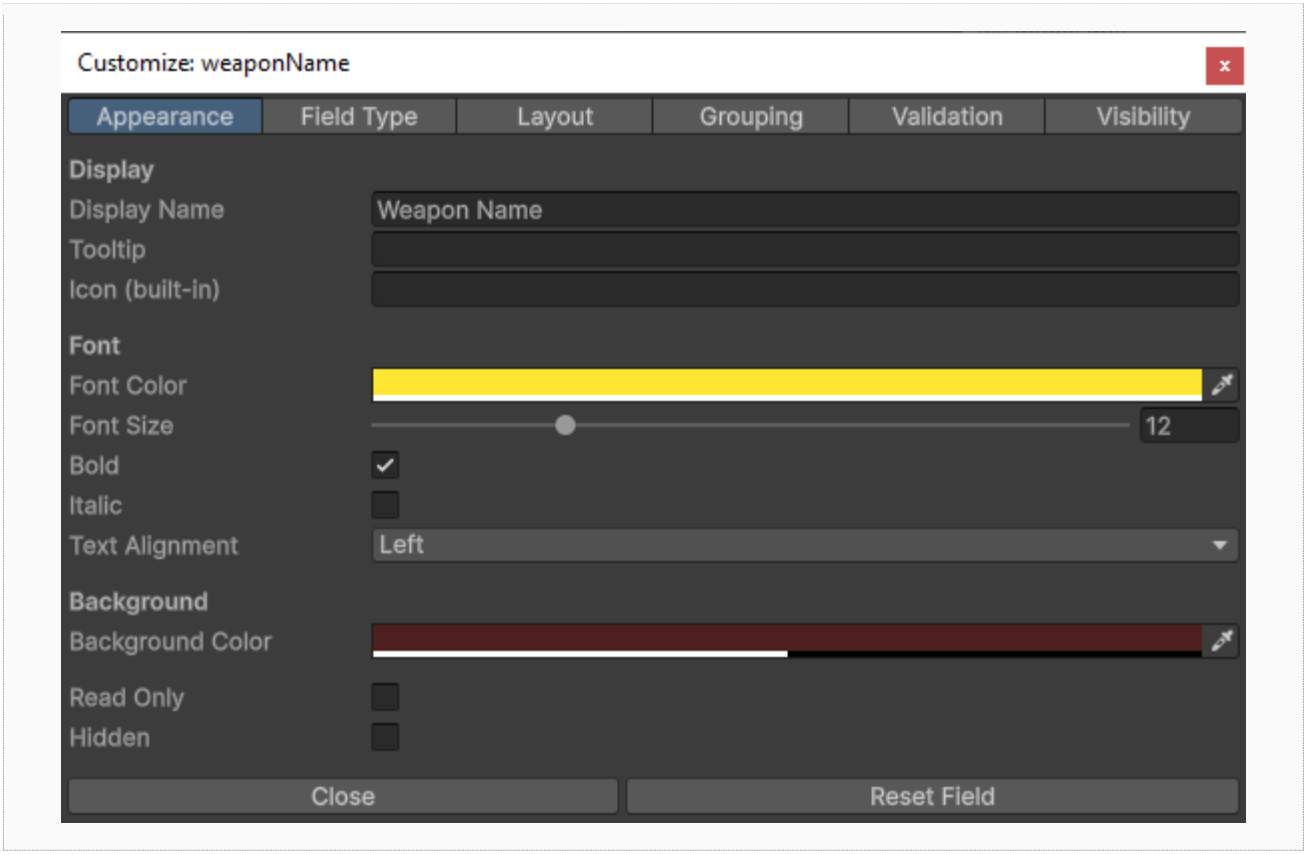


Fig 6.2 — Appearance tab showing font color, size, bold/italic, and background color settings

Field Type Tab

Override the field's rendering type. The dropdown shows only compatible types based on the property's data type. When you select a type, additional configuration options appear:

- **Slider —** Min/max values and fill/track/thumb colors
- **Min-Max Slider —** Lower/upper values and range limit
- **Progress Bar —** Color, background color, and label format string
- **Knob —** Min/max values
- **Button Tabs —** Custom tab labels, active/inactive background and text colors
- **Image Preview —** Preview height in pixels
- **Dropdown —** Background and text color

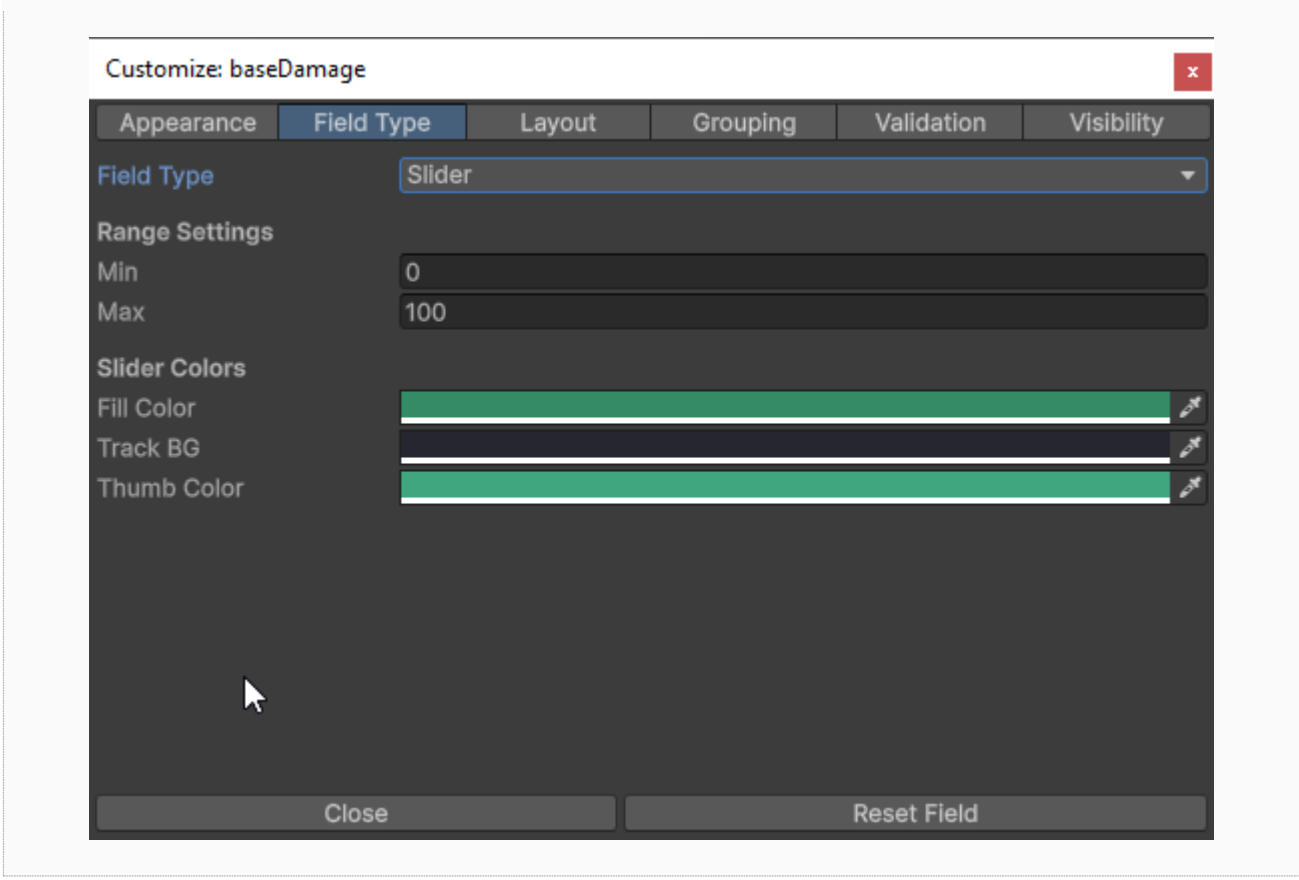


Fig 6.3 — Field Type tab with Slider selected, showing min/max, fill color, track color, and thumb color

Conditions Tab

Set up conditional visibility — show or hide this field based on another field's value.

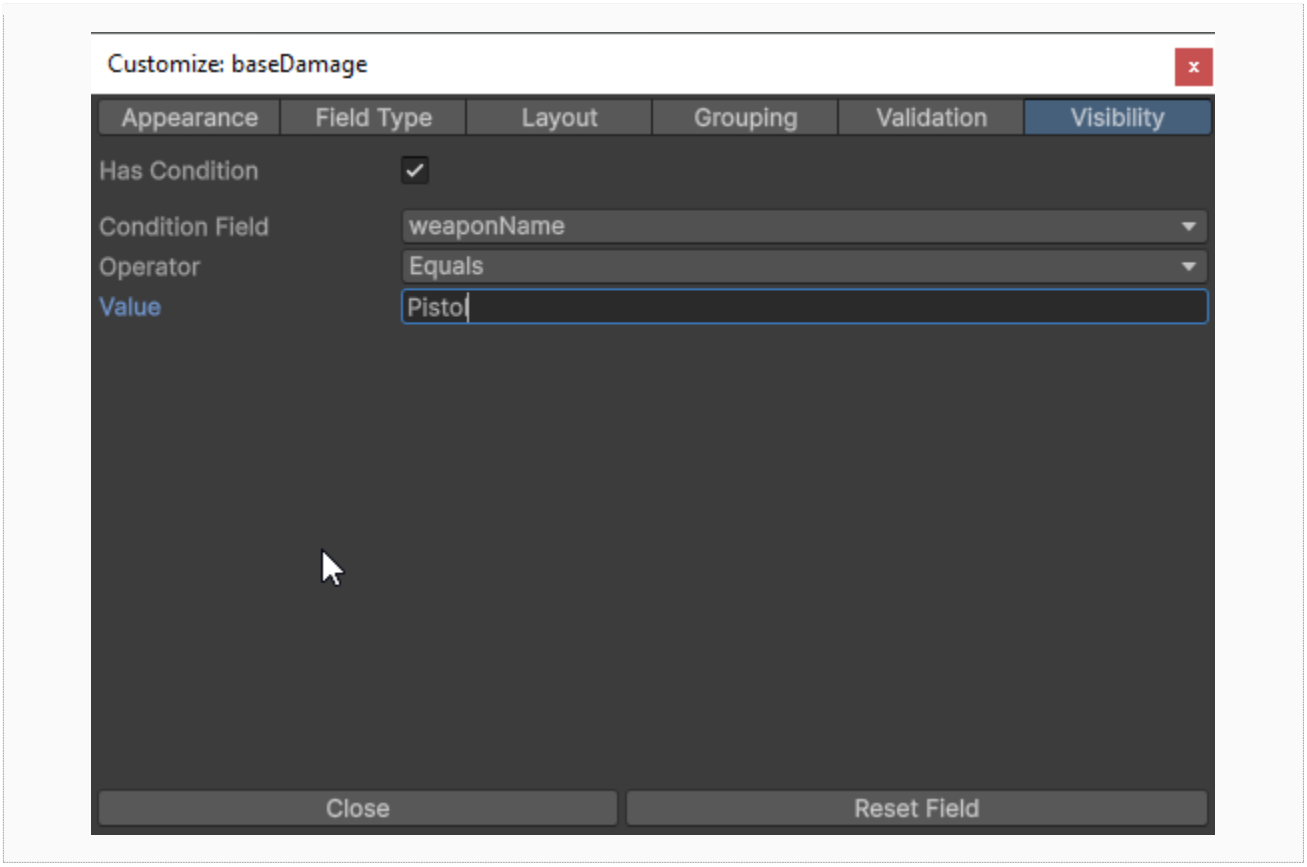


Fig 6.4 — Conditions tab showing field-based conditional visibility setup

Validation Tab

Add validation rules with custom error messages.

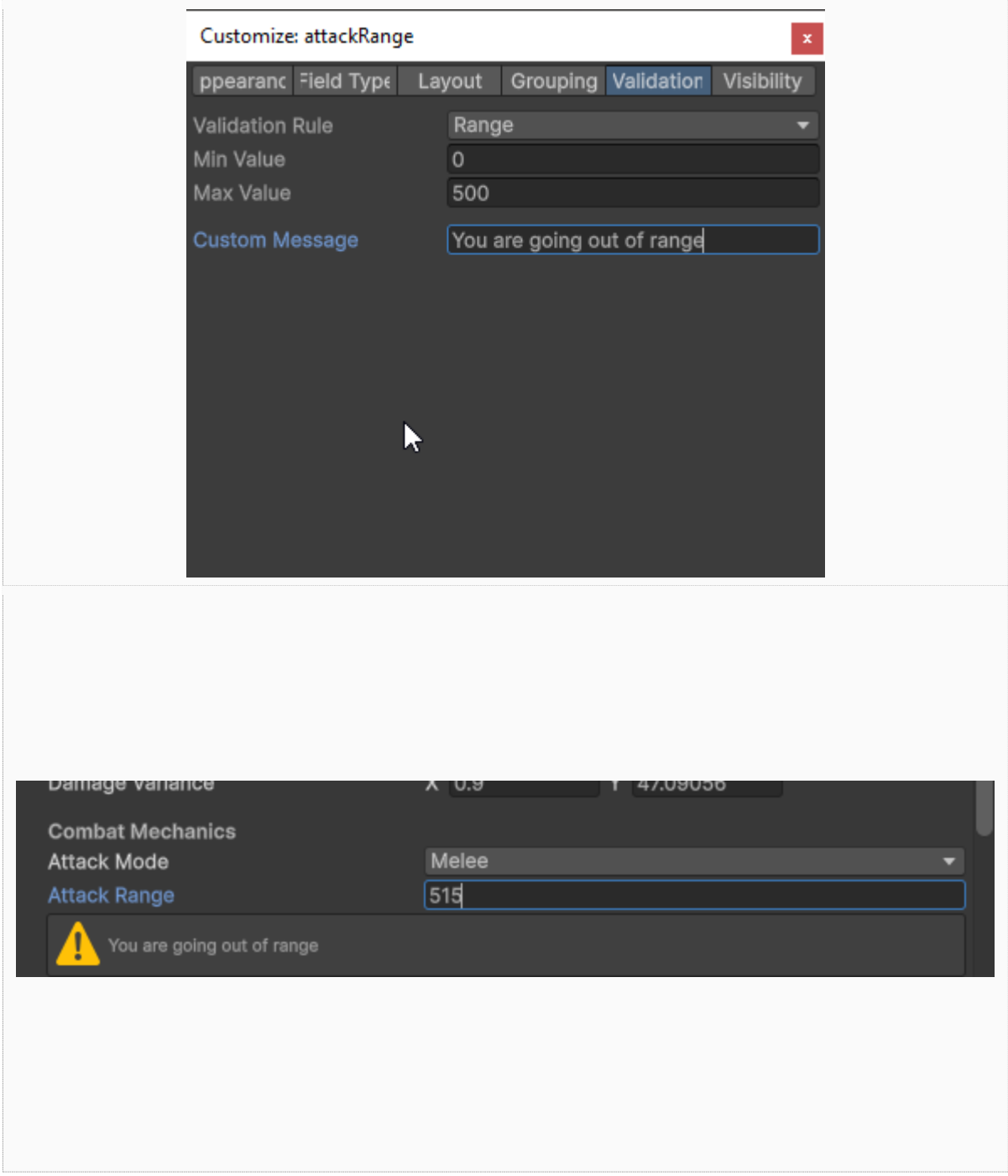


Fig 6.5 — Validation tab showing Range validation with min/max and error message

Layout Tab

Manage headers and layout elements that appear above this field. You can add headers with icons, separators, and spaces.

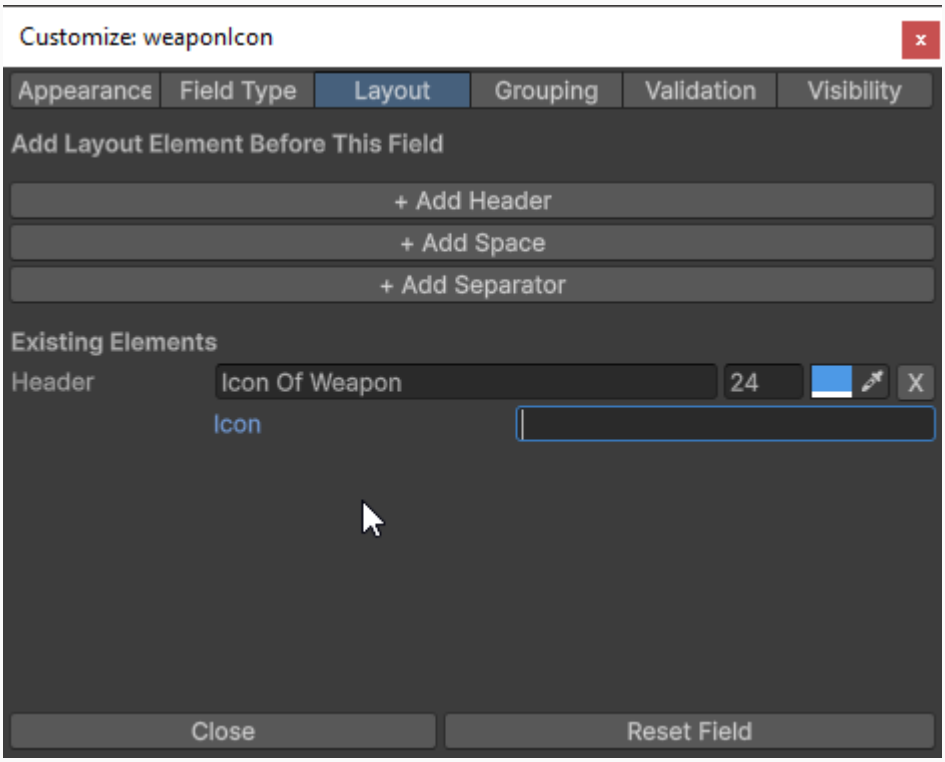


Fig 6.6 — Layout tab showing a header element with icon field and color settings

6. Designer Window

The Designer Window is the main editor window for managing all aspects of a profile. Open it via Tools > Inspector Designer Pro > Designer Window

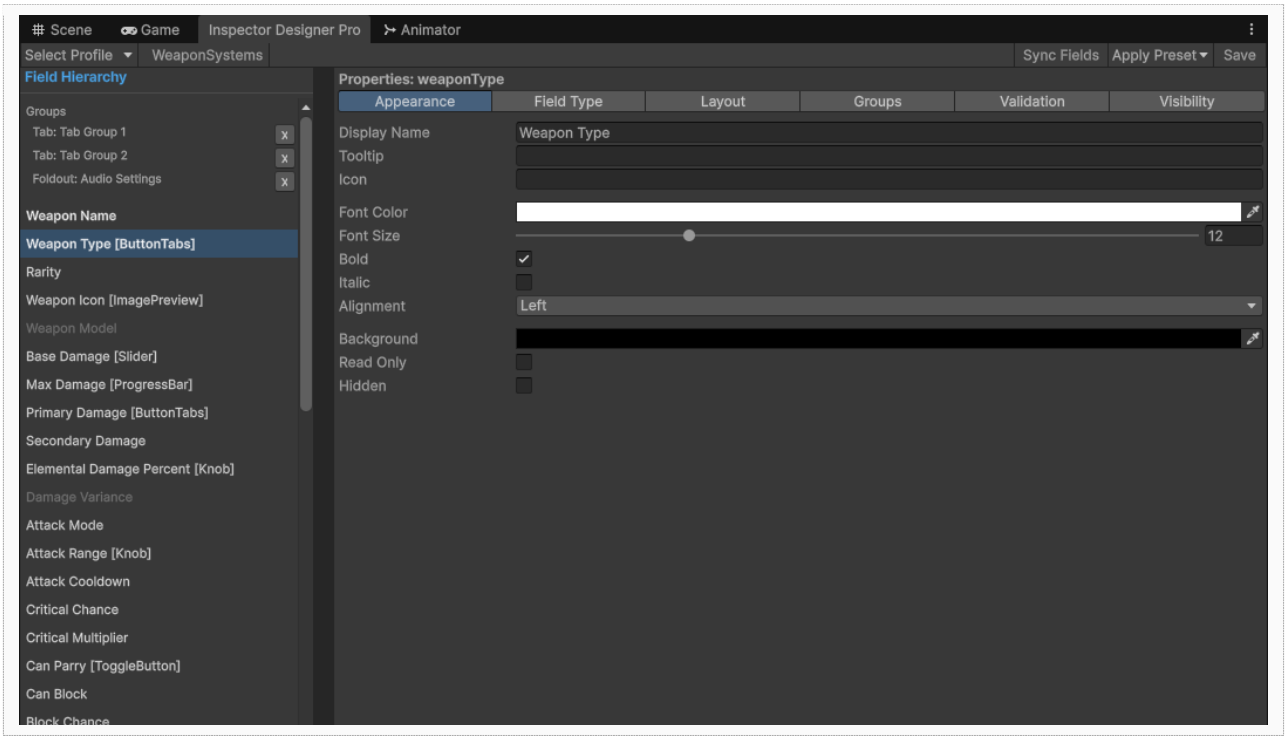


Fig 7.1 — The Designer Window showing the full interface

Selecting a Profile

At the top of the window, select the MonoBehaviour type you want to customize. If no profile exists yet, one will be created automatically when you make your first change.

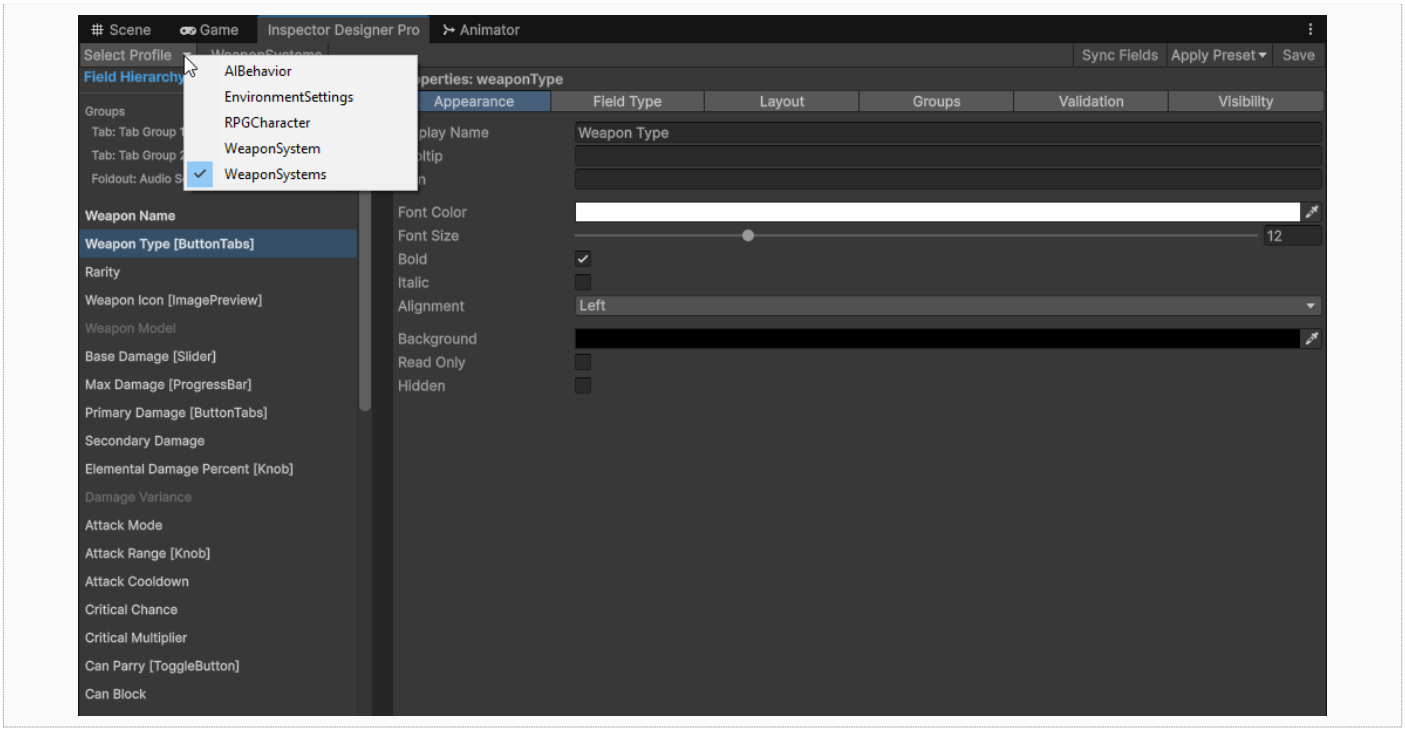


Fig 7.2 — Profile selector dropdown at the top of the Designer Window

Window Tabs

Global Tab

Set defaults that apply to the entire inspector:

- Inspector background color
- Default font color for all fields
- Default font size
- Spacing between fields

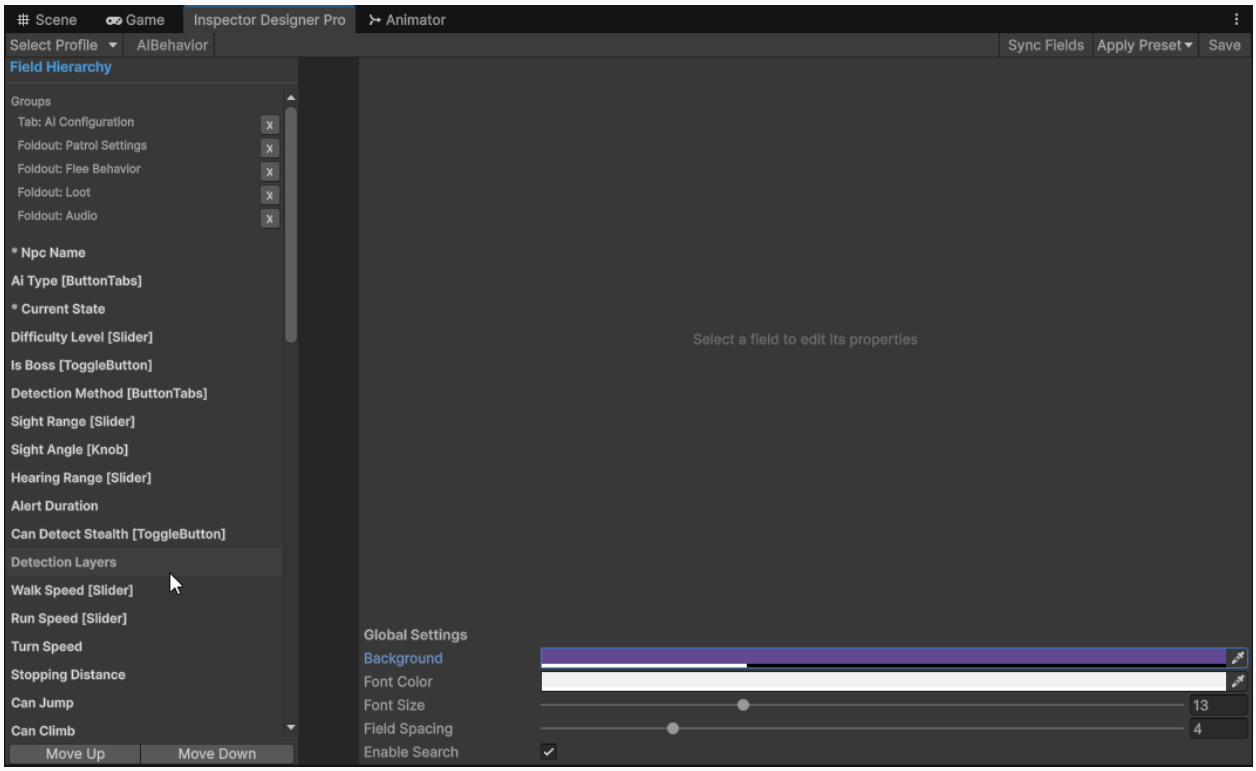


Fig 7.3 — Global tab showing background color, font color, font size, and field spacing

Fields Tab

Lists every serialized field on the component. For each field, you can configure:

- Display name and tooltip
- Font color, size, bold, italic
- Background color
- Field type override with type-specific settings
- Read-only and hidden toggles

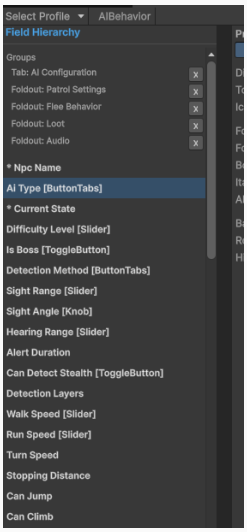


Fig 7.4 — Fields tab with multiple fields showing their individual settings

Layout Tab

Add and manage layout elements — headers, separators, spaces, and box groups. Each element can be positioned before any field and styled independently.

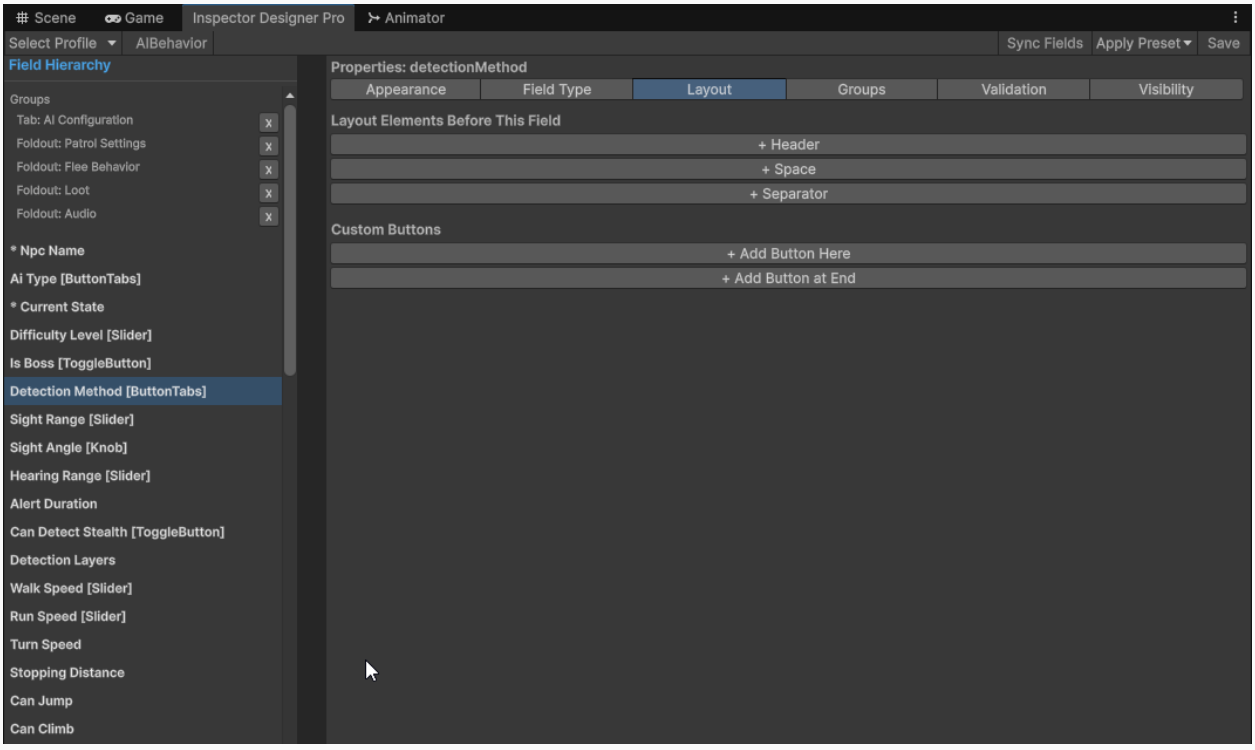


Fig 7.5 — Layout tab showing headers, separators, and spaces with position and styling options

Groups Tab

Create and manage Tab Groups and Foldout Groups. This is where you configure group names, which fields belong to each group, and all group styling options.

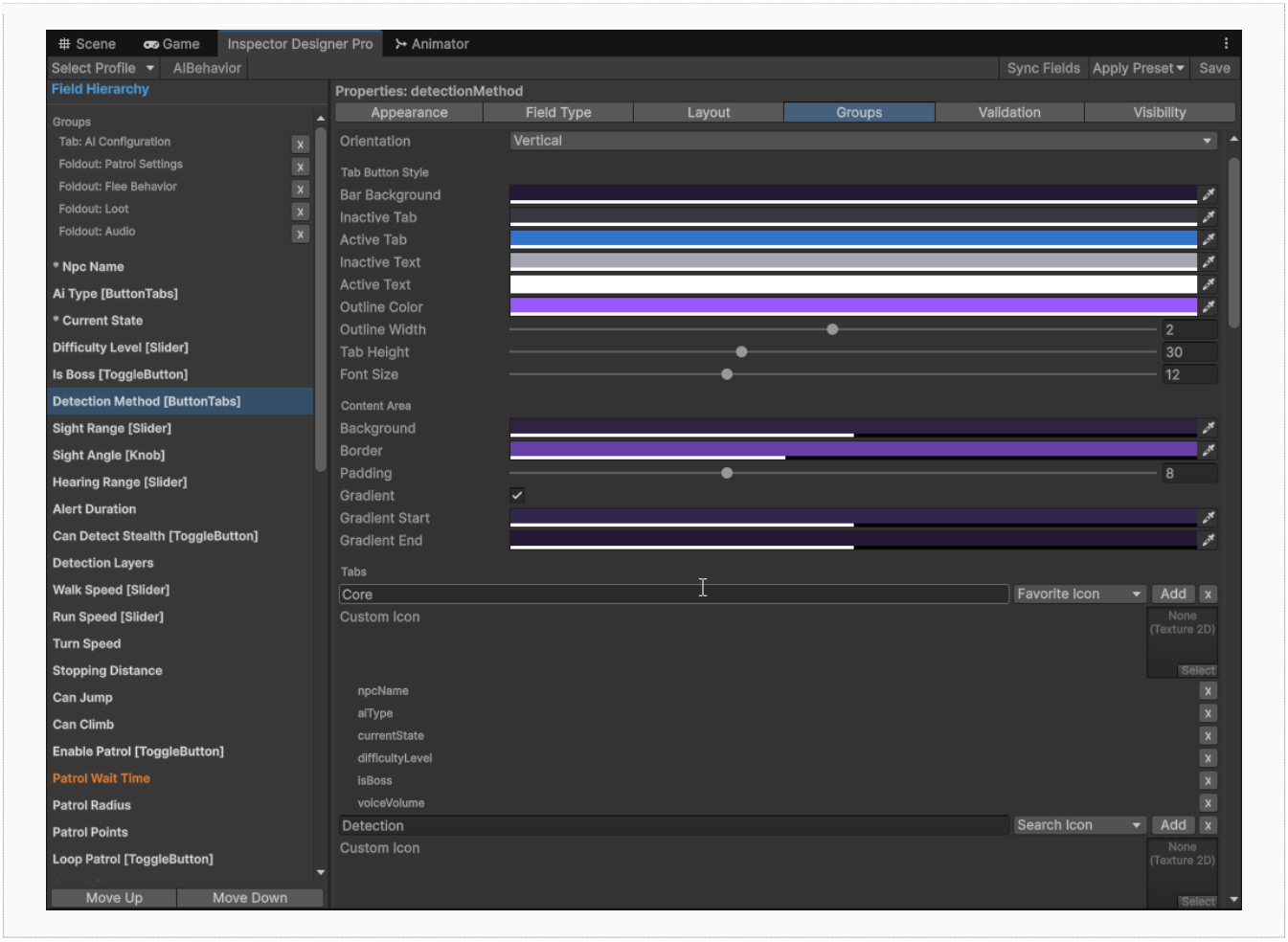


Fig 7.6 — Groups tab showing a tab group with styling options expanded

7. Field Type Overrides

Inspector Designer Pro provides 12 field type overrides that change how a property renders in the inspector. Each override is designed for specific property types — the UI automatically filters to show only compatible options.

Slider

Converts a numeric field (int or float) into a horizontal slider with min/max range. Fully stylable with custom colors for the fill bar, background track, and drag thumb.

- Min Value — The minimum value of the slider range
- Max Value — The maximum value of the slider range
- Fill Color — Color of the filled portion of the slider
- Track Color — Background color of the slider track
- Thumb Color — Color of the draggable handle

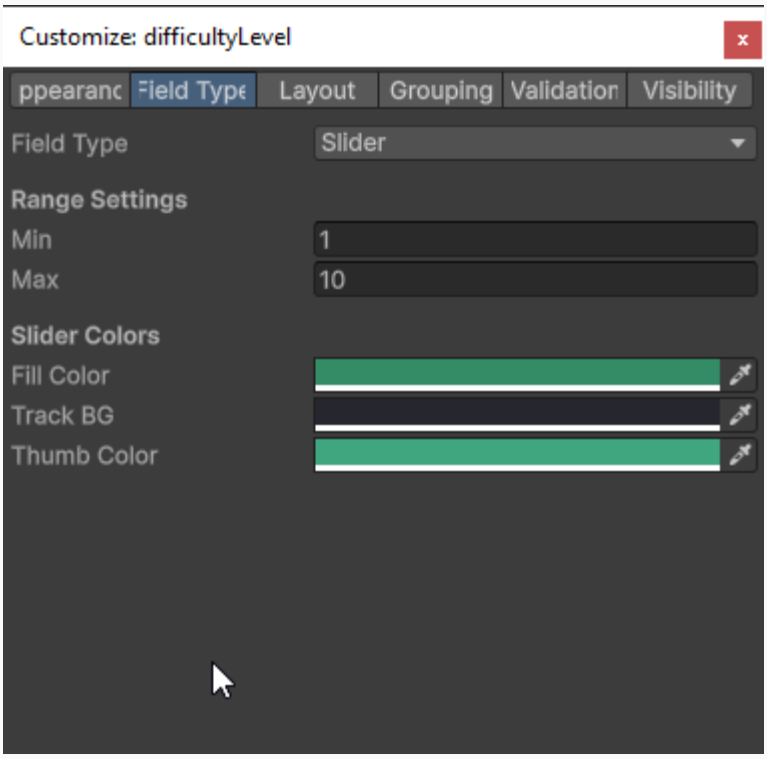


Fig 8.1 — A float field displayed as a styled slider with custom fill and thumb colors

Min-Max Slider

A dual-handle range slider for selecting a range between two values. Useful for random ranges (e.g., spawn delay between 0.5 and 2.0 seconds).

- Lower / Upper — The current selected range
- Limit — The maximum allowed range

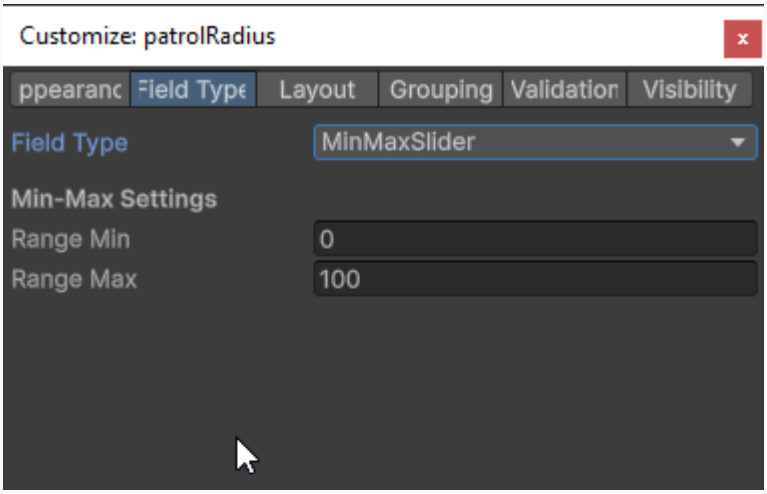


Fig 8.2 — A min-max slider showing a selected range

Progress Bar

Displays a numeric value as a visual progress bar. Read-only display ideal for health, mana, loading progress, or any 0-to-max value.

- Bar Color — Fill color of the progress bar
- Background Color — Track color behind the bar
- Label Format — Printf-style format string (default: "{0:P0}" for percentage)

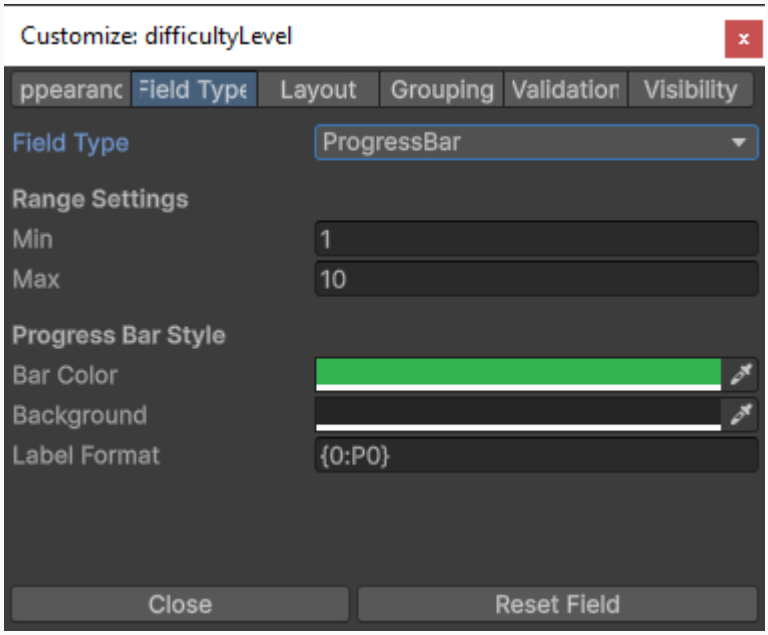


Fig 8.3 — A health value displayed as a green progress bar with percentage label

Knob

A rotary dial control for numeric values. Provides an intuitive, compact way to adjust values by rotating.

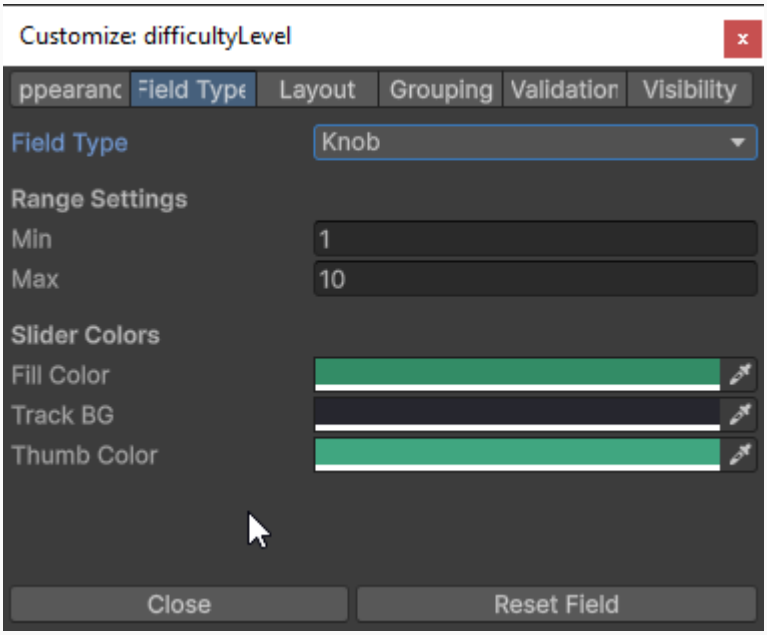


Fig 8.4 — A knob control for adjusting a rotation angle value

Percentage Field

Displays a float value (0.0 to 1.0) as a percentage (0% to 100%). The underlying value stays as a float for code use, but appears as a percentage in the inspector.

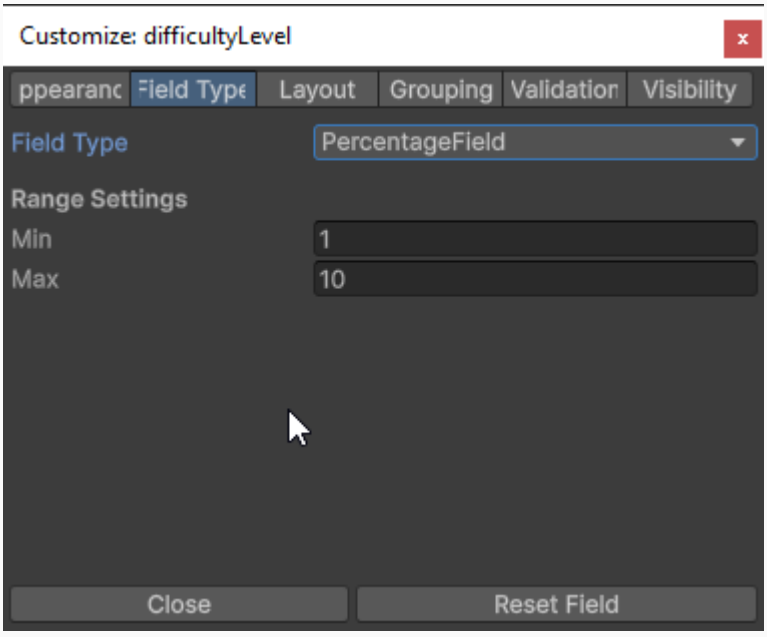


Fig 8.6 — A float field displayed as a percentage

Toggle Button

Replaces the default checkbox with a stylized on/off toggle switch button. Provides a cleaner, more visible boolean toggle.

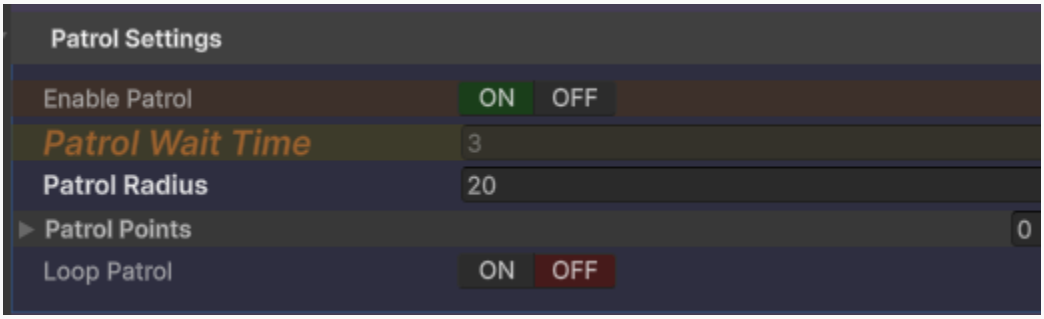


Fig 8.7 — A boolean field displayed as a toggle button

Dropdown

A styled enum popup with custom background and text colors. When you change the background color, Inspector Designer Pro automatically draws a custom dropdown arrow triangle to ensure it remains visible regardless of color choice.

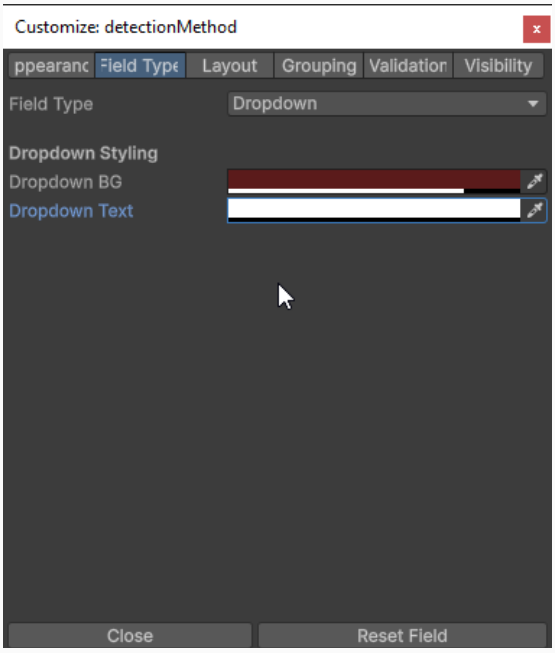


Fig 8.8 — An enum field with custom dark blue dropdown background and white text

Button Tabs

Converts an enum or integer into a toolbar-style tab selector. Each option appears as a clickable tab button instead of a dropdown. Fully stylable with active/inactive colors for both background and text.

- Active Background Color — Background of the selected tab
- Inactive Background Color — Background of unselected tabs
- Active Text Color — Text color of the selected tab
- Inactive Text Color — Text color of unselected tabs
- Custom Labels — Override the display text for each tab option

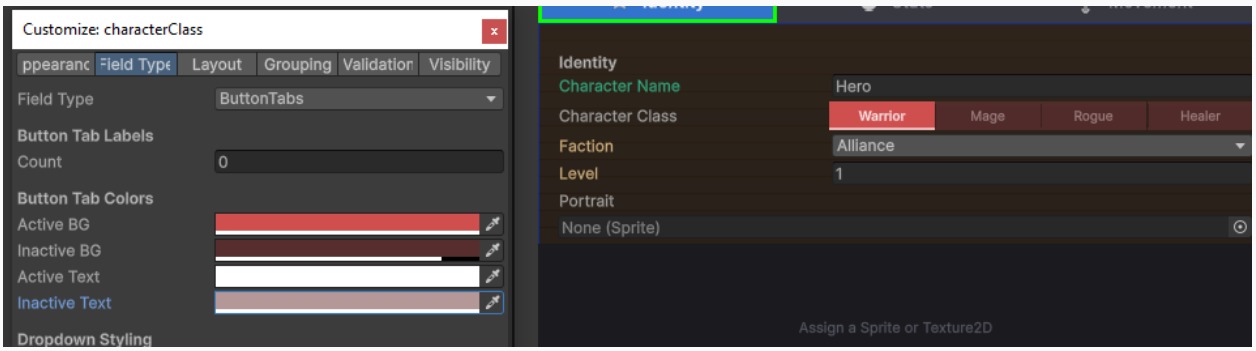


Image Preview

Displays a Texture2D or Sprite field as an inline image preview below the object field. The preview height is adjustable.

- Preview Height — Height in pixels for the image preview area



Fig 8.11 — A sprite field with inline image preview showing the texture

8. Tab Groups

Tab Groups let you organize fields into a tabbed interface. Only the fields in the currently selected tab are visible, keeping the inspector clean and navigable.

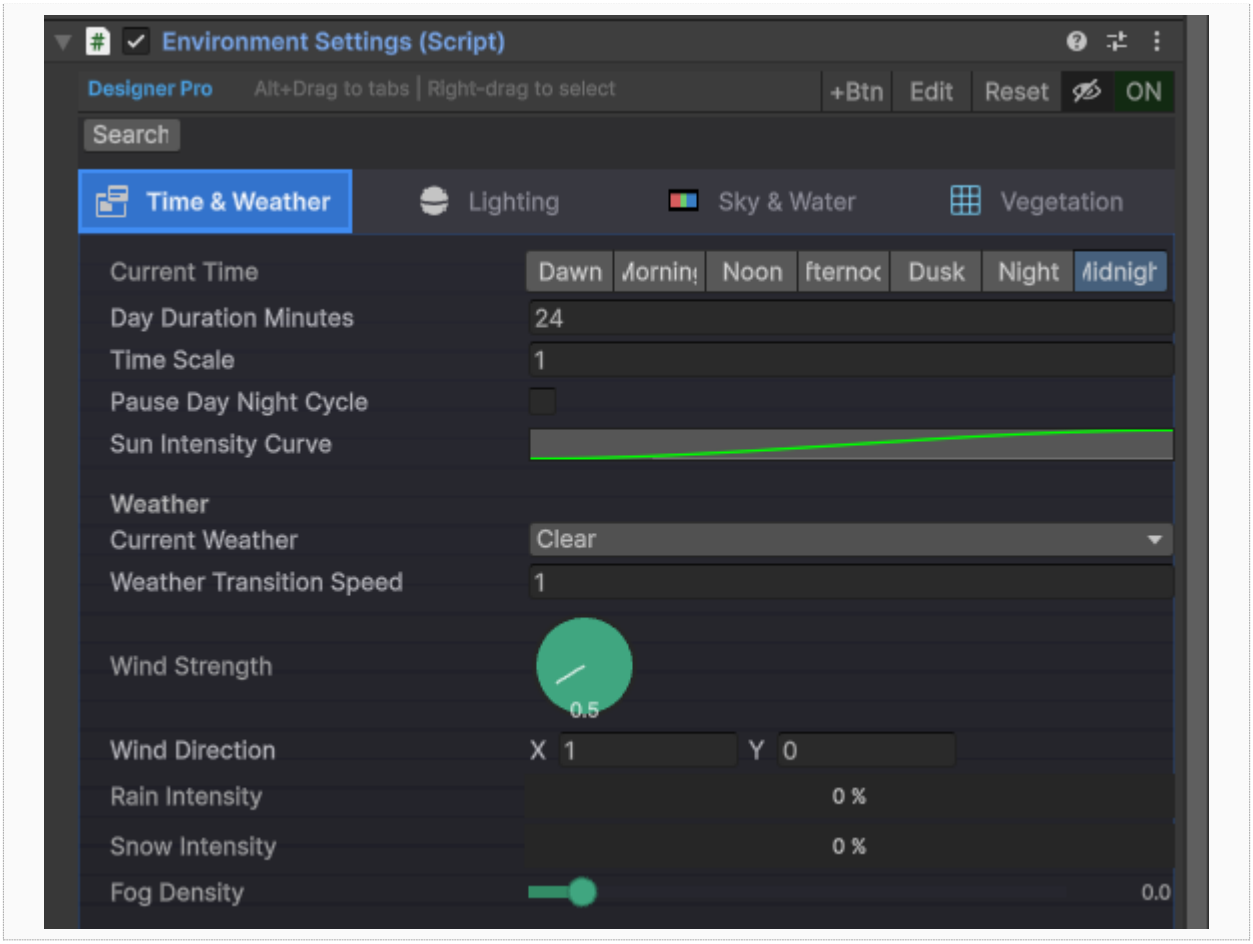


Fig 10.1 — A tab group with four tabs: Time & Weather,Lightning, Sky&Water and Visuals

Creating a Tab Group

5. Right-click any field and choose Add to Tab Group > New Tab Group.
6. Enter a name for the group and the first tab.
7. Right-click more fields and add them to the same group under different tabs.
8. Alternatively, use the Designer Window Groups tab to create and manage everything.

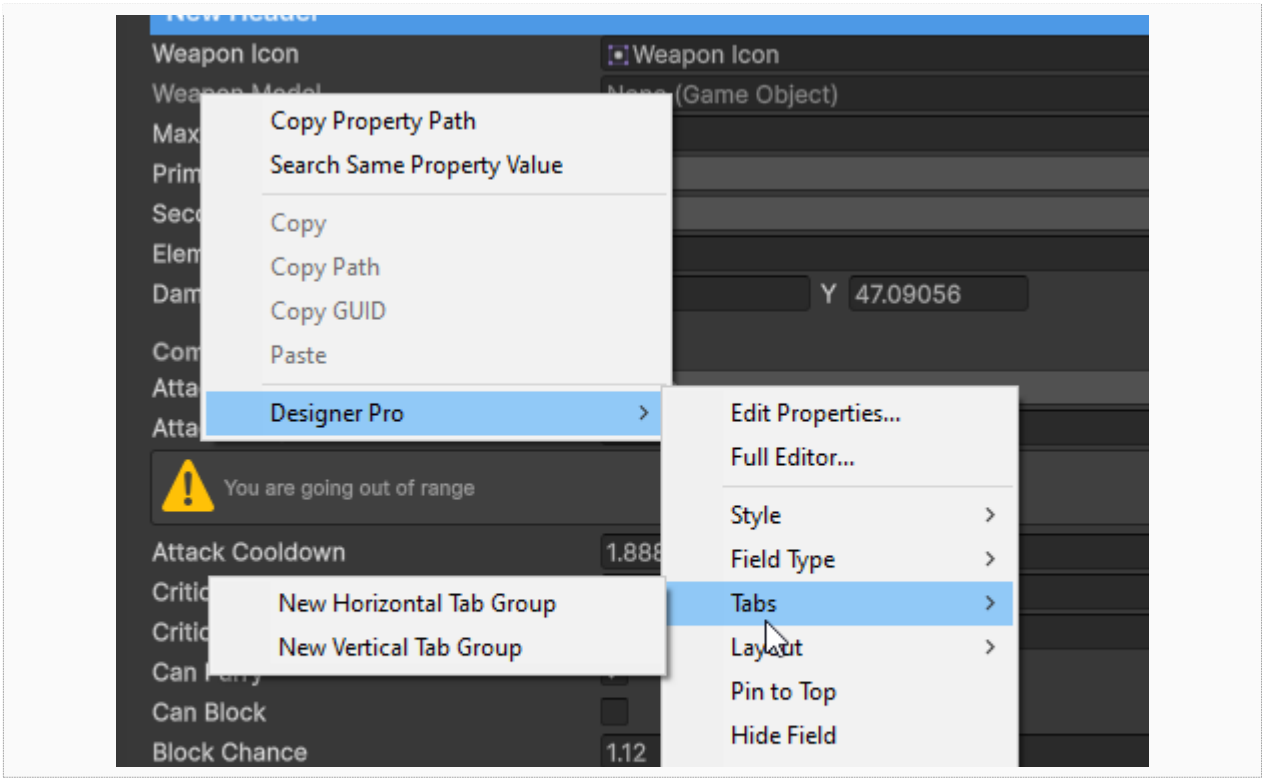


Fig 10.2 — Creating a new tab group from the right-click context menu

Tab Orientation Modes

Two layout modes for tab buttons:

Horizontal Tabs

Standard tabs arranged in a row across the top. Best for 2-5 tabs with text labels.

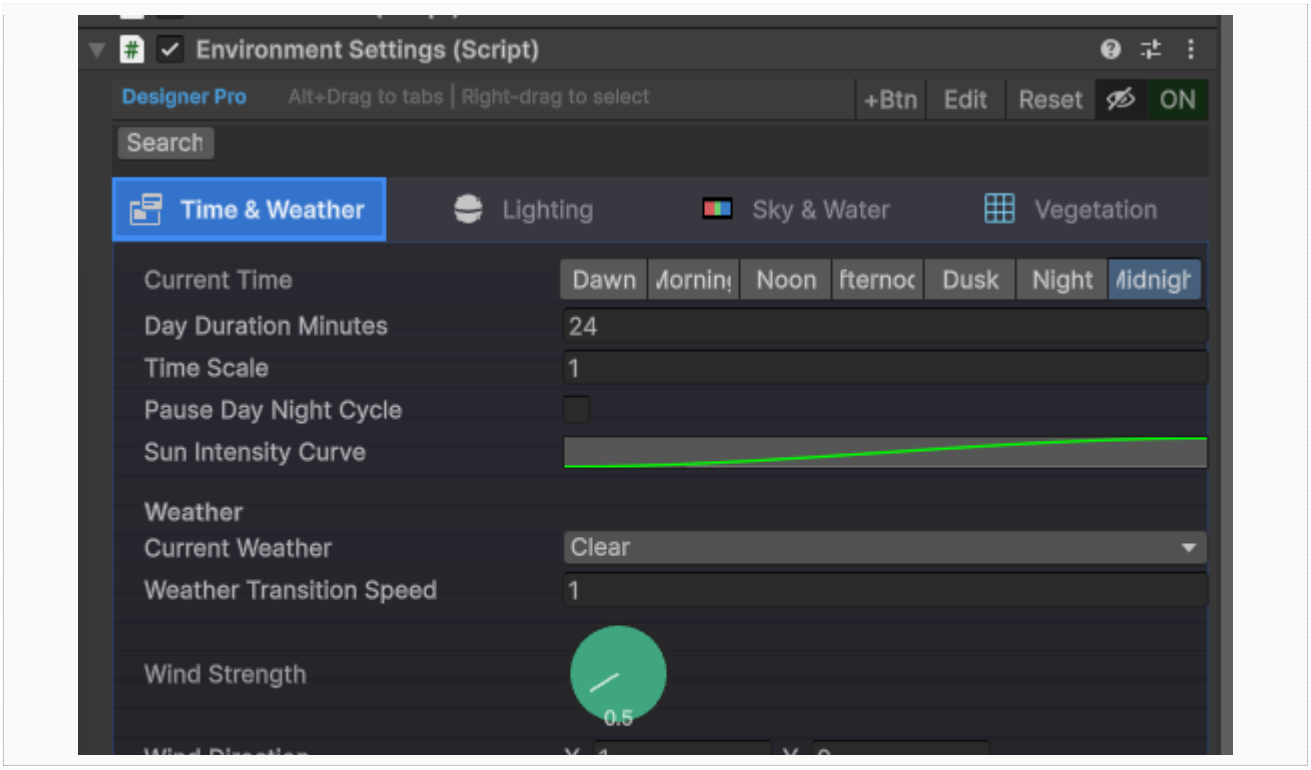


Fig 10.3 — Horizontal tab layout with text labels

Vertical Tabs

Sidebar-style tabs stacked vertically on the left side. Good for many tabs or longer tab names.

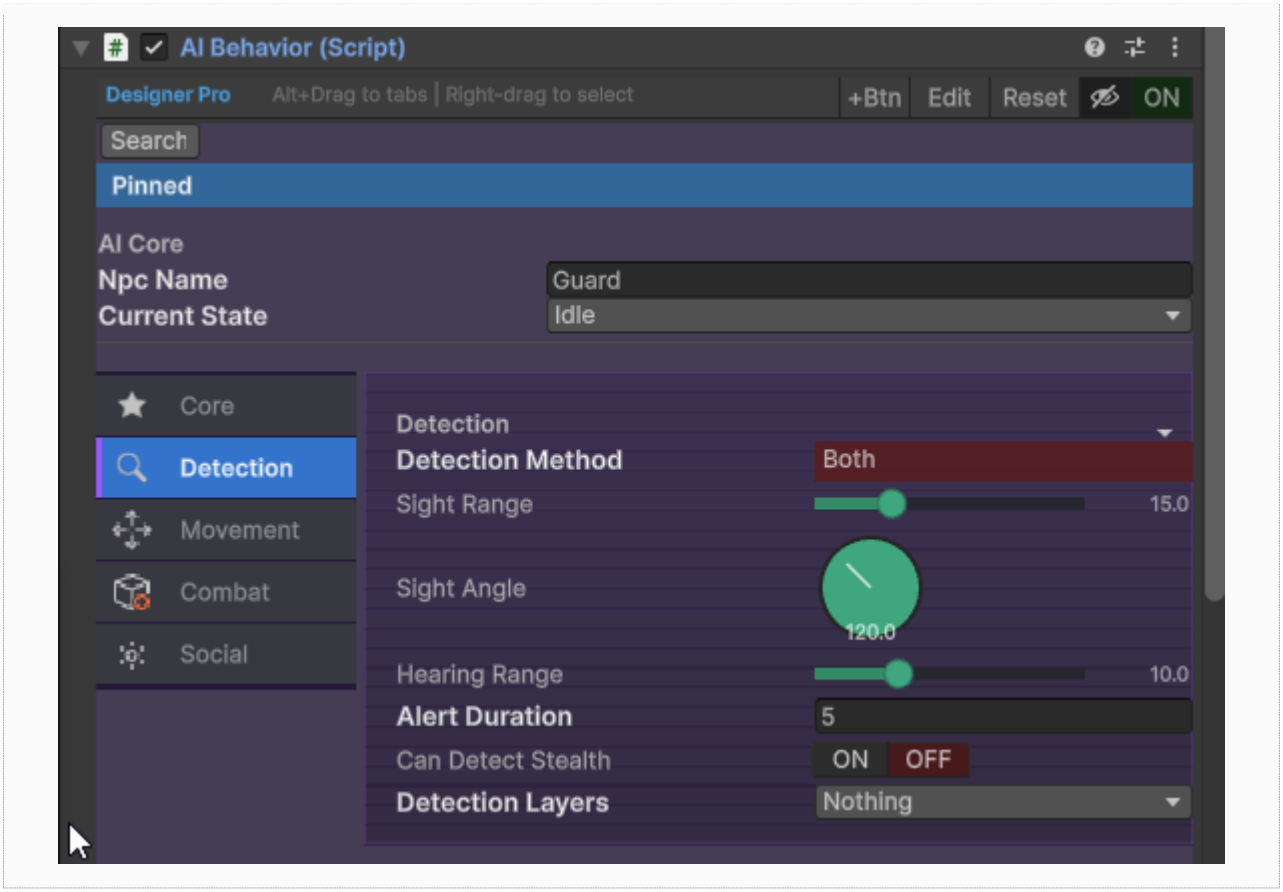


Fig 10.4 — Vertical tab layout as a sidebar

Tab Styling Options

Every visual detail of tab groups is customizable:

Property	Description
Tab Color	Background color of inactive tabs
Active Tab Color	Background color of the currently selected tab
Tab Text Color	Text color of inactive tab labels
Active Tab Text Color	Text color of the selected tab label
Tab Bar Background	Background color of the entire tab bar area
Selected Outline Color	Outline/border color around the active tab
Selected Outline Width	Thickness of the active tab outline in pixels
Tab Height	Height of tab buttons in pixels
Tab Font Size	Font size of tab labels
Content Background	Background color of the tab content area
Content Border Color	Border color around the content area
Content Border Width	Border thickness in pixels

Content Padding	Inner padding of the content area in pixels
Use Gradient	Enable gradient background for the content area
Gradient Start / End	Start and end colors when gradient is enabled

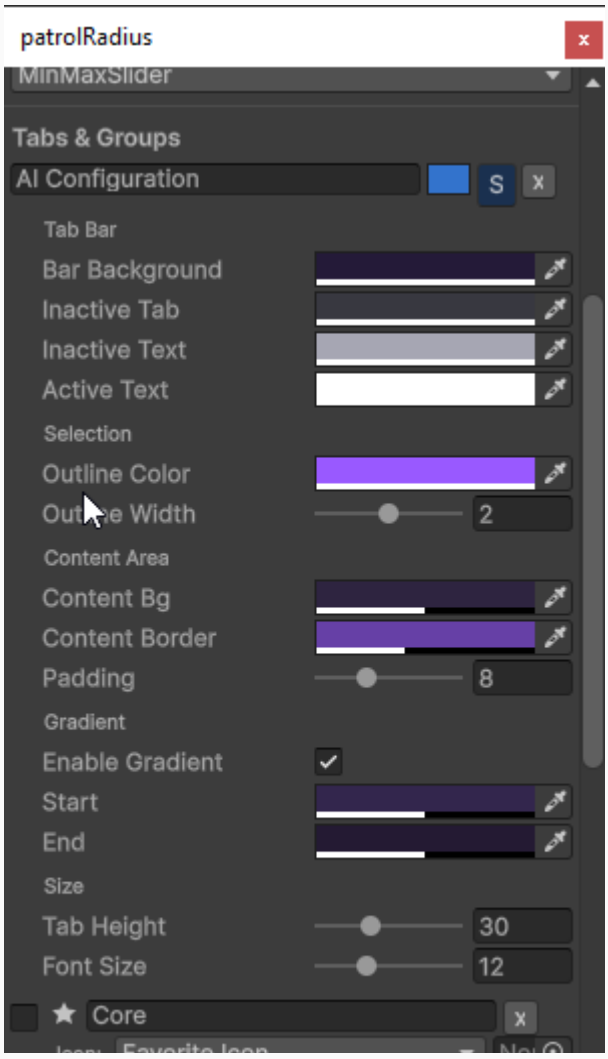


Fig 10.6 — Tab styling options in the Designer Window

Per-Tab Overrides

Each individual tab can override the group's content background with its own color or gradient. This lets you visually distinguish different tab sections.

- Per-tab content background color
- Per-tab gradient toggle with custom start and end colors
- Per-tab icon (Unity built-in icon name or custom Texture2D)

9. Foldout Groups

Foldout Groups create collapsible sections in the inspector. Click the header to expand or collapse the group. Useful for organizing related fields that don't need to be visible all the time.

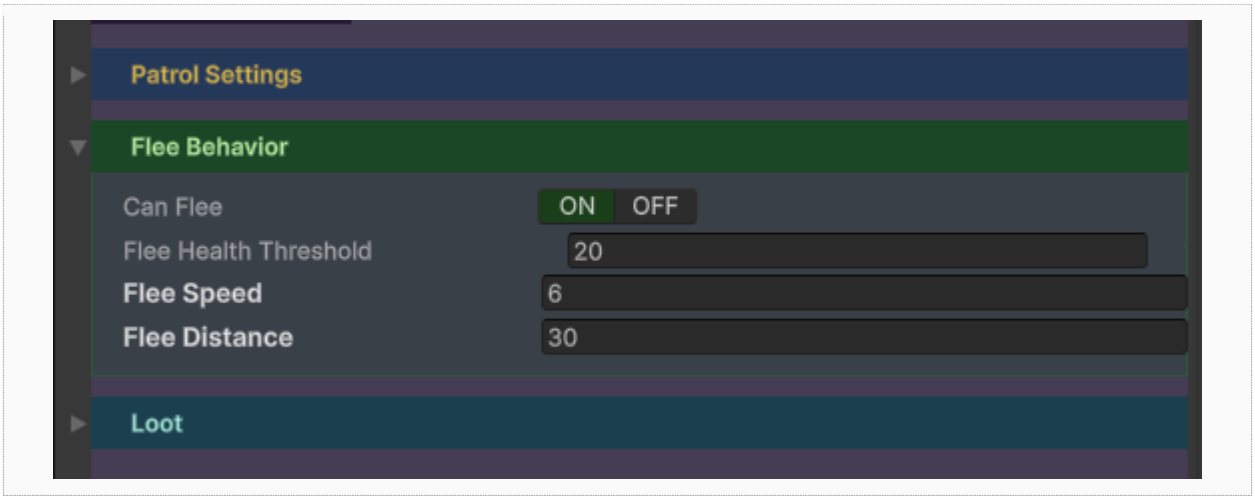


Fig 11.1 — Multiple foldout groups, one expanded and one collapsed

Creating a Foldout Group

9. Right-click a field and choose Designer Pro > Layout> New Foldout Group
10. Enter a name for the foldout.
11. Add more fields to the same foldout from the context menu.

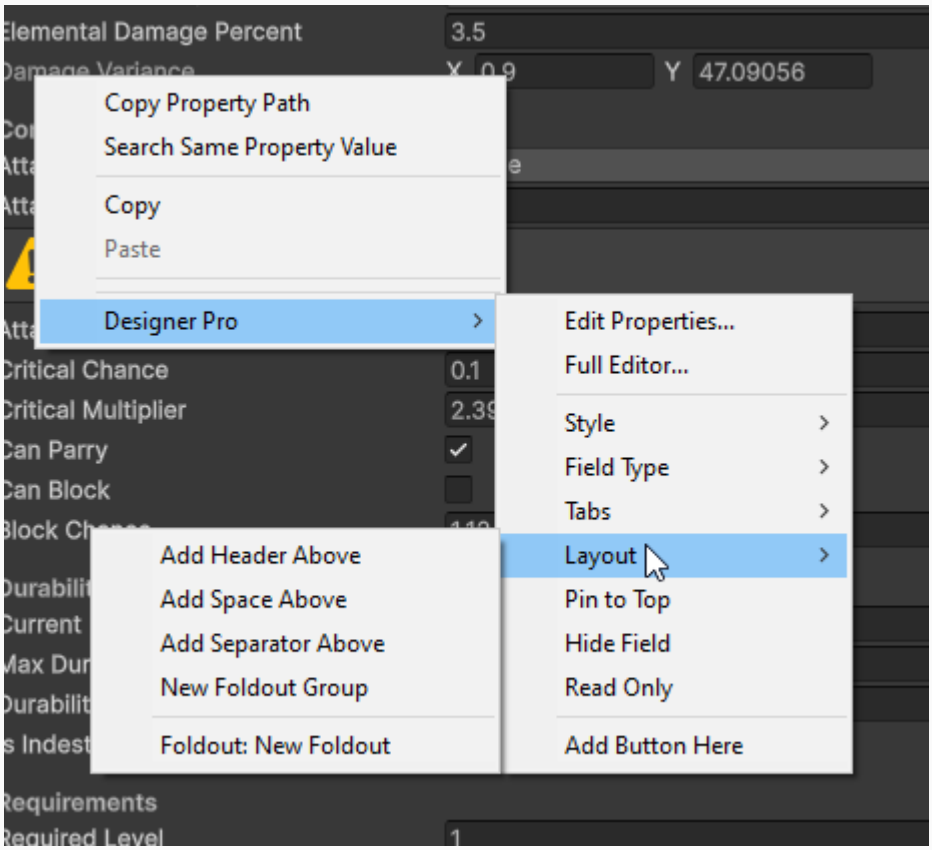


Fig 11.2 — Adding fields to a foldout group via right-click menu

Foldout Styling

Complete visual control over both the header and content area:

Header Styling

- Background color
- Text color
- Font size
- Bold toggle
- Icon — select from Unity's built-in icon library via the icon picker popup

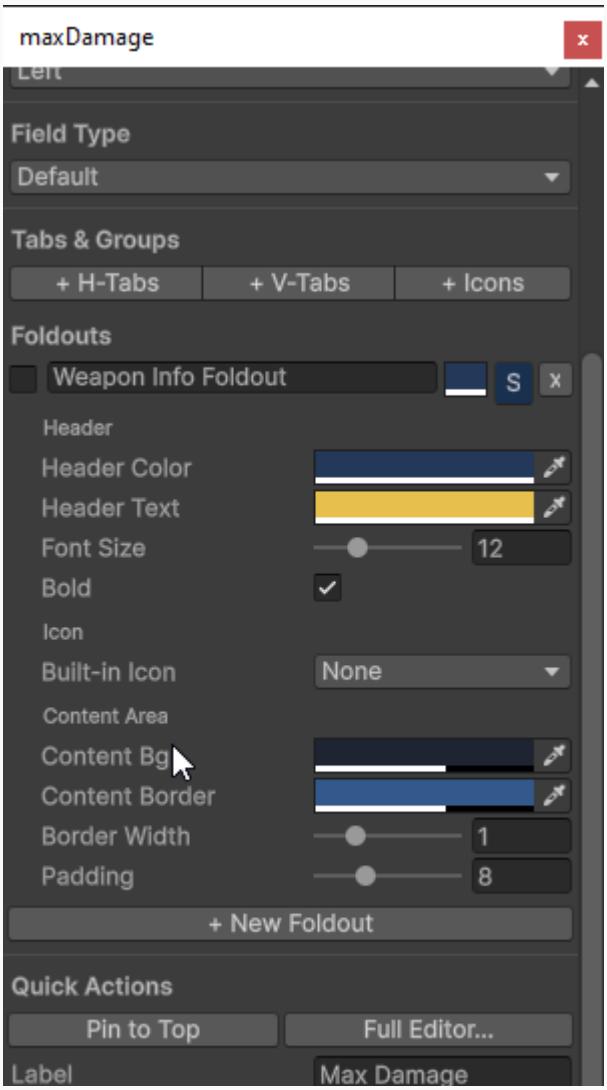


Fig 11.3 — Foldout headers with custom colors and icons

Content Area Styling

- Content background color
- Content border color
- Content border width (in pixels)
- Content padding (inner spacing in pixels)

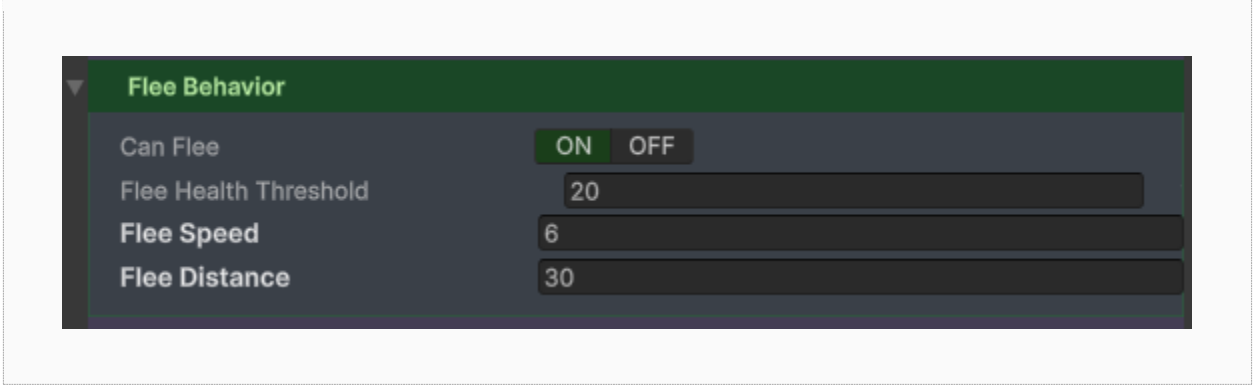


Fig 11.4 — A foldout with styled content area: dark background, blue border, generous padding

Additional Options

- Default expanded/collapsed state
- Column position: full width, left column, or right column (for side-by-side layouts)

10. Custom Buttons

Add clickable buttons to your inspector that invoke public methods on your MonoBehaviour at edit time. Useful for editor utilities like "Reset Values", "Generate Data", "Randomize", etc.

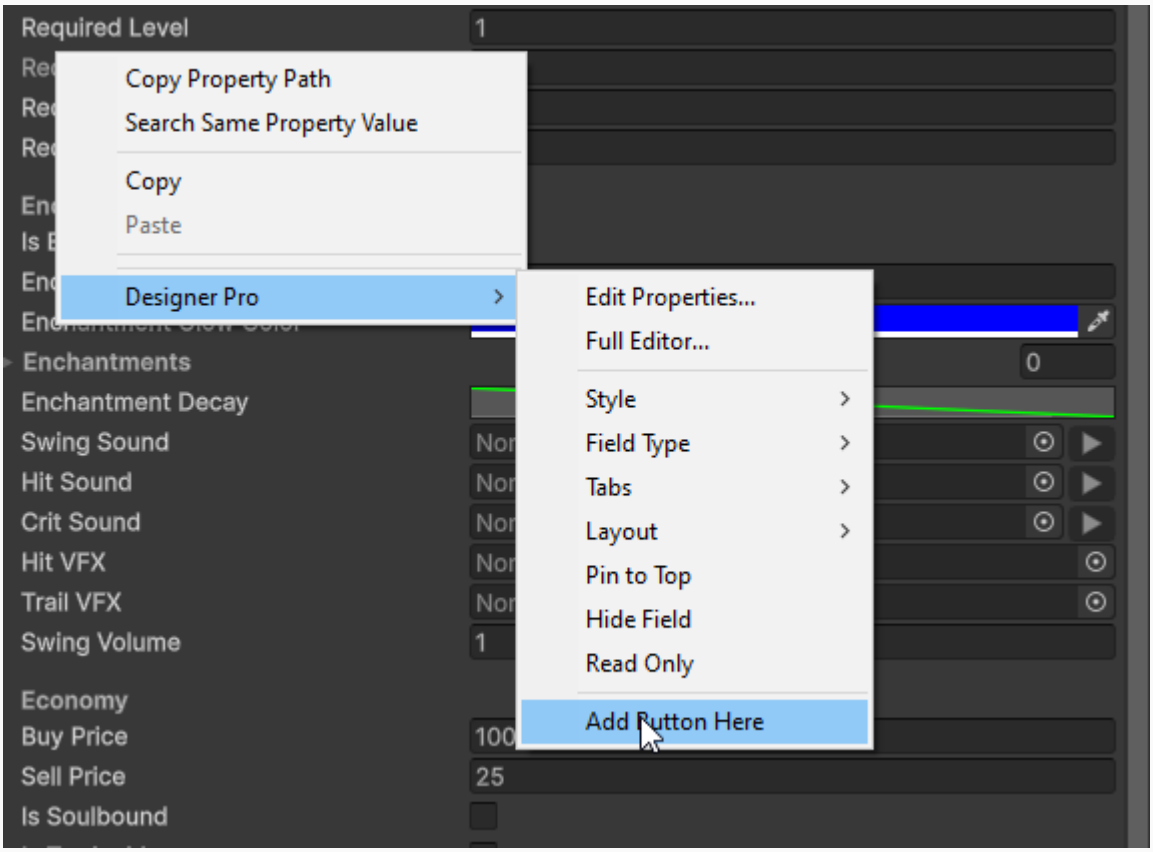


Fig 13.1 — Creating Custom Button With Right Click Context Menu

Button Properties

Property	Description
Button Label	The text displayed on the button
Method Name	The name of a public method on the MonoBehaviour to call
Background Color	Button background color
Text Color	Button text color
Outline Color	Border/outline color
Outline Width	Border thickness in pixels
Corner Radius	Rounded corner radius in pixels (0 = sharp corners)

Width	Button width in pixels (0 = full width)
Height	Button height in pixels
Font Size	Font size of the button label
Bold	Bold toggle for the label
Position	Insert before a specific field, or leave empty for bottom of inspector

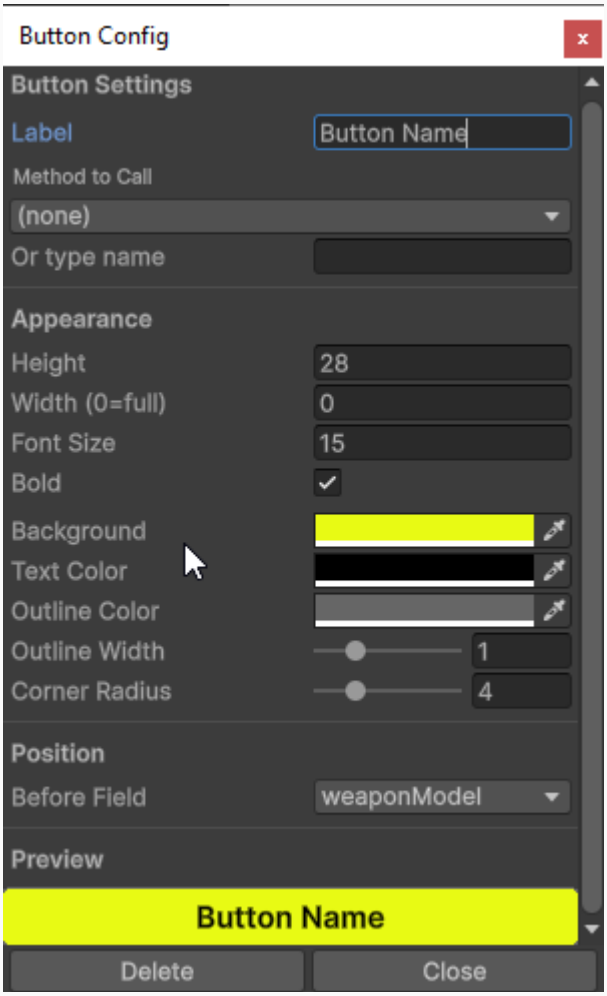


Fig 13.2 — Button configuration in the Designer Window

Tip: The method must be public and take no parameters. It is called using reflection at edit time.

11. Marquee Multi-Select & Bulk Actions

Hold right-click and drag across multiple fields to select them all at once. Selected fields are highlighted with a blue overlay.

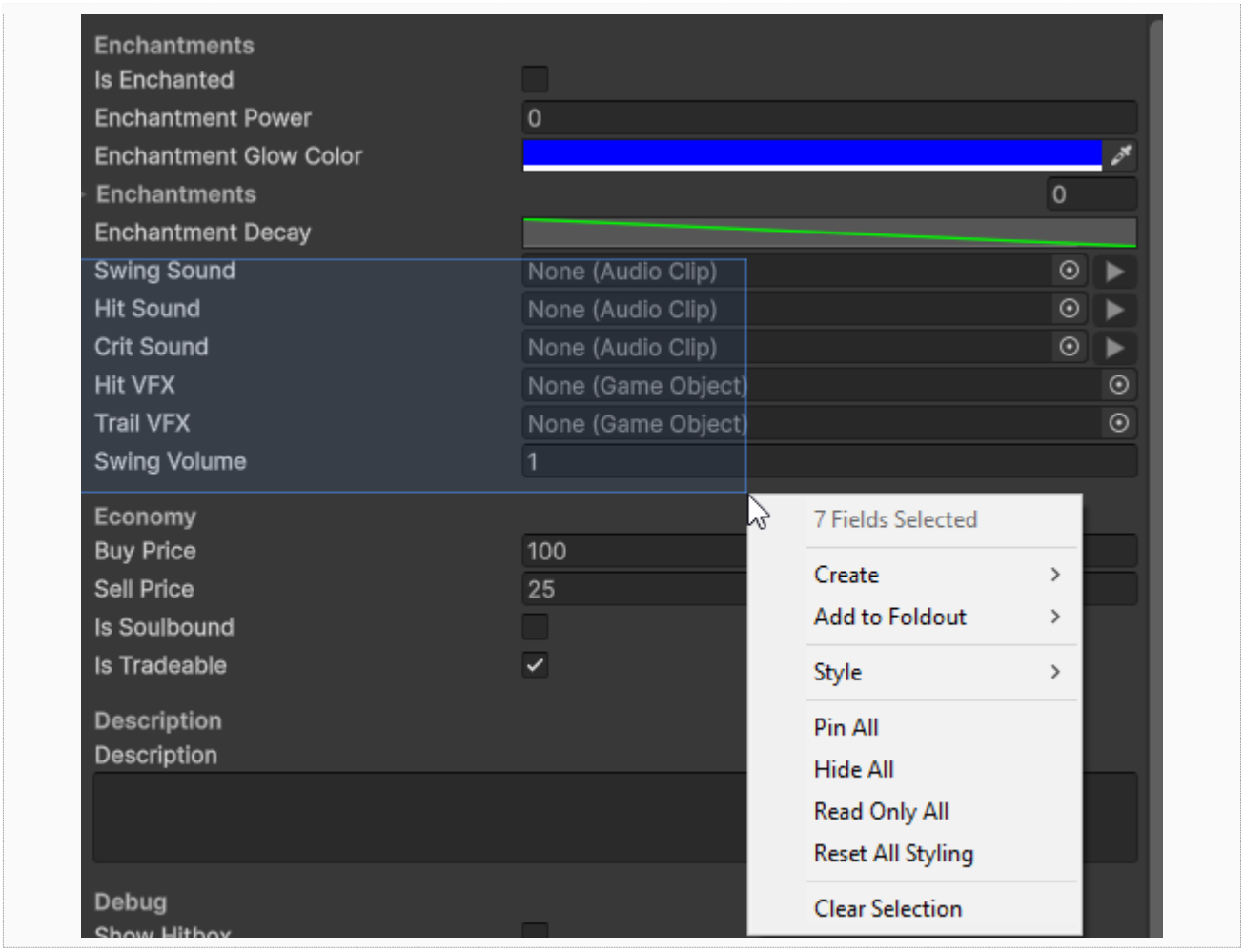


Fig 20.1 — Multiple fields selected with the marquee highlight

Bulk Actions

Once multiple fields are selected, right-click to apply actions to all of them simultaneously:

- Add all selected fields to a tab group
- Add all selected fields to a foldout group
- Change field type for all selected fields
- Pin or hide all selected fields
- Apply styling (colors, font) to all selected fields

This is extremely useful when organizing a component with many fields — select 10 fields at once and add them all to a tab in one action.

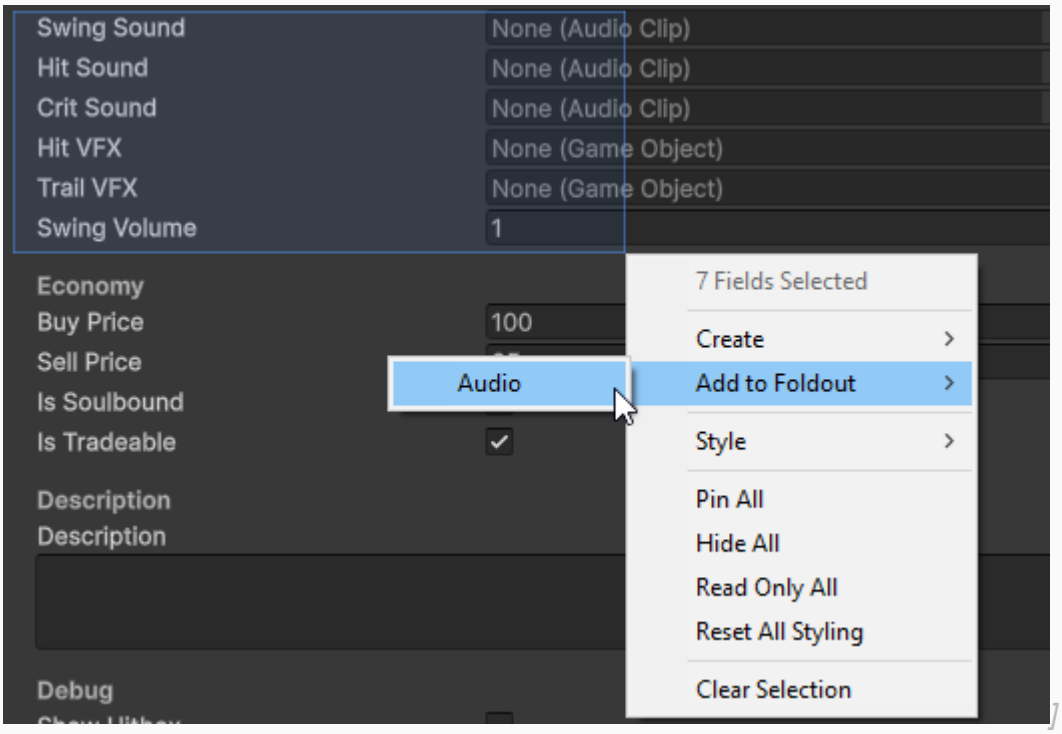


Fig 20.2 — Bulk right-click menu appearing for multiple selected fields

> **Note:** Left-clicking anywhere clears the selection without accidentally toggling checkboxes or changing field values.

12. Search & Pinned Fields

Search

Components with many serialized fields can be hard to navigate. The built-in search bar (when enabled) appears at the top of the inspector and filters fields by name as you type.

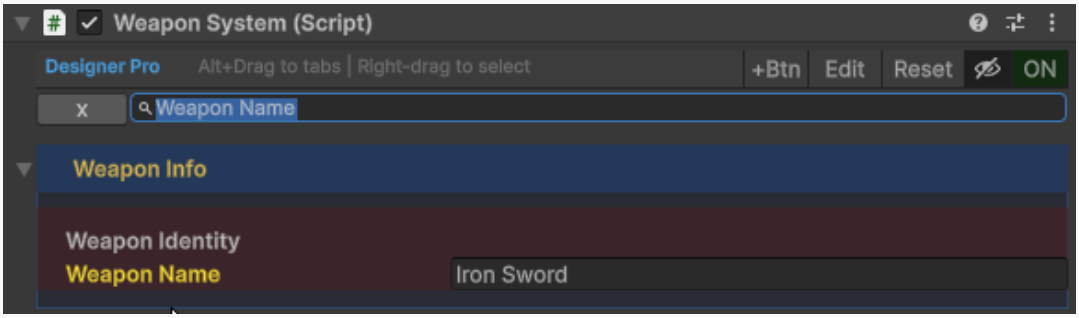


Fig 21.1 — Search bar filtering fields by keyword

Pinned Fields

Pin your most important fields to always appear at the top of the inspector, regardless of grouping or ordering. Pinned fields stay visible even when other fields are hidden by search or tab selection.

Pin a field via right-click > Pin to Top.

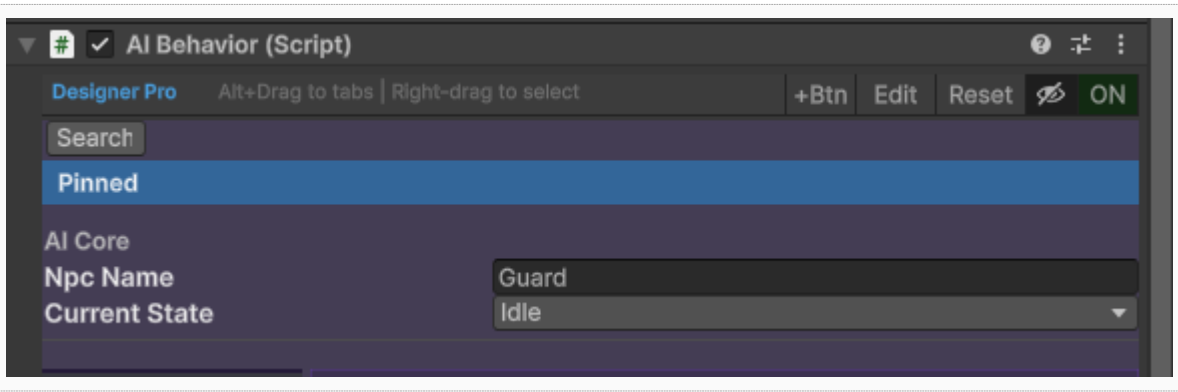


Fig 21.2 — Pinned fields showing at the top of the inspector above grouped content

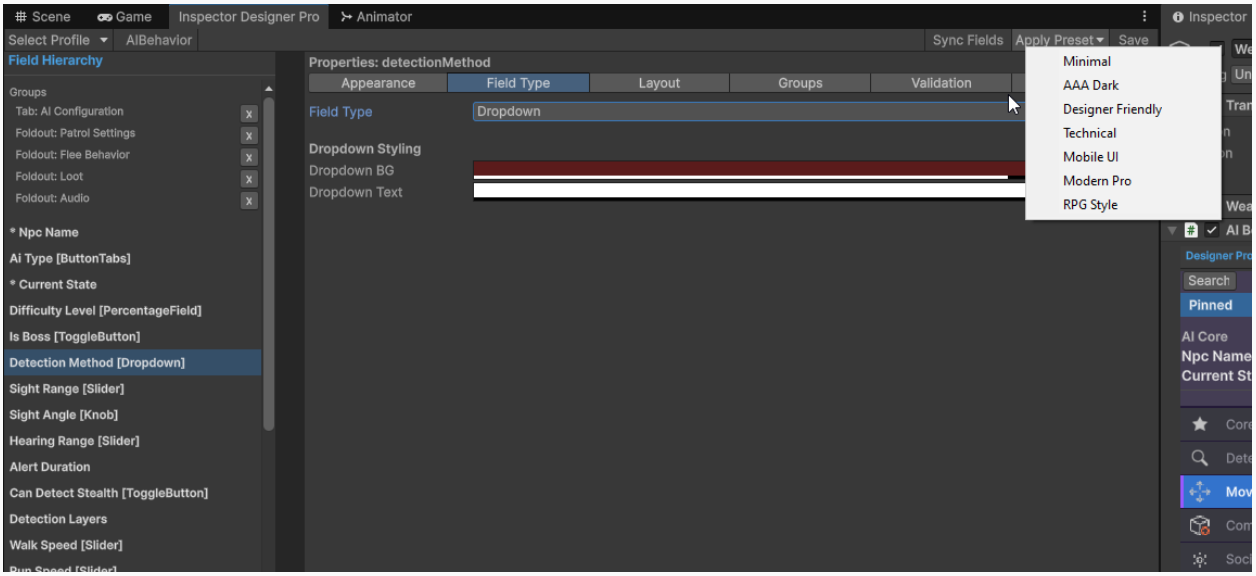
13. Preset Themes

7 professionally designed themes are included. Apply any preset with one click from the Designer Window or right-click menu, then customize further.

Preset	Style

Minimal	Clean, neutral grays. Small fonts. No visual clutter.
AAA Dark	Deep blue tones on dark backgrounds. Professional studio look.
Designer Friendly	Purple accent colors. Large fonts. Generous spacing.
Technical	Green-on-dark terminal aesthetic. Compact and data-dense.
Mobile UI	Bright blue accents. Large touch-friendly sizing. High contrast.
Modern Pro	Subtle blue accents on dark gray. Clean and contemporary.
RPG Style	Warm gold and brown tones. Fantasy/medieval theme.

Each preset configures: global background/font colors, tab group colors (active, inactive, bar, outline, content), foldout header and text colors, and layout element styling.



> **Tip:** Presets are a starting point. After applying one, customize individual fields, groups, and colors to match your project's style.

14. Undo Support

Every modification you make through Inspector Designer Pro is fully undoable with Ctrl+Z (Windows) or Cmd+Z (Mac).

Over 80 individual undo recording points cover every possible action:

- Changing field display names, tooltips, colors, fonts, bold, italic
- Changing field types and type-specific settings (slider ranges, progress bar colors, etc.)
- Adding/removing fields from tab groups and foldout groups
- Creating, renaming, or deleting groups
- Reordering fields within groups
- Adding/removing layout elements (headers, separators, spaces)
- Adding/removing custom buttons
- Changing global styling
- Applying presets
- Pinning, hiding, and setting read-only state
- Changing validation rules and conditions
- Renaming tabs, foldouts, and headers via right-click

> **Note:** Undo uses Unity's built-in Undo.RecordObject system, so it integrates seamlessly with Unity's Edit > Undo menu and history.

15. Profile System

All customization in Inspector Designer Pro is stored in ScriptableObject profiles — standard Unity assets that live in your project.

How Profiles Work

- One profile per MonoBehaviour type (identified by class name).
- Profiles are created automatically the first time you customize a field.
- Stored as .asset files in your project's Assets folder.
- Profiles contain all field metadata, groups, layout elements, buttons, and styling.

Version Control

Profiles are standard Unity ScriptableObject assets. They work seamlessly with all version control systems:

- Git (with .meta files)
- Perforce
- PlasticSCM / Unity Version Control
- Any system that handles Unity .asset and .meta files

Team Sharing

Since profiles are project assets, they are shared automatically when the project is shared. All team members see the same customized inspectors.

> **Tip:** Profiles are independent of scenes — they apply project-wide to any instance of the MonoBehaviour type.

16. Technical Details

Compatibility

- Unity 2021.3 LTS and newer (all LTS and Tech Stream versions)
- Compatible with Unity 6 and all current Unity versions
- Works with all render pipelines: Built-in, URP, and HDRP
- All target platforms — the package is editor-only and excluded from builds

Zero Build Impact

This is a critical feature for production projects:

- The entire package is editor-only (Assembly Definition with includePlatforms: ["Editor"])
- No runtime scripts or assemblies are included in builds
- Zero increase in build size
- Zero risk of build failures
- Zero runtime performance impact
- Safe to include in any project regardless of target platform (PC, Mobile, Console, WebGL, VR)

Architecture

- Built on Unity's IMGUI system using EditorWindow, SerializedObject, and SerializedProperty
- ScriptableObject-based profile system for data persistence
- Modular field drawer architecture: FieldDrawerBase abstract class with FieldDrawerRegistry for type discovery
- AudioClip preview uses reflection on Unity's internal AudioUtil class with safe try/catch fallbacks

Performance

Inspector Designer Pro is designed for zero editor lag:

- No per-frame texture allocations — uses a cached Dictionary<int, Texture2D> for color textures
- Rounded button corners rendered via Handles.DrawSolidDisc (no texture generation or GC pressure)
- Custom dropdown arrows via Handles.DrawAAConvexPolygon (zero allocation per frame)
- Style overrides use temporary GUIStyle modifications, restored immediately after drawing
- Lightweight IMGUI drawing with minimal overhead

Package Contents

- ~35 editor-only C# scripts organized in Editor/ subdirectory

- 7 built-in preset themes
- 4 demo MonoBehaviour scripts for testing (RPGCharacter, WeaponSystem, EnvironmentSettings, AIBehavior)
- Built-in Unity icon library integration
- Clean namespace: InspectorDesignerPro
- Single Assembly Definition scoped to Editor platform
- No third-party dependencies

17. Troubleshooting & FAQ

Q: I customized my inspector but changes don't appear.

Make sure you have a GameObject selected with the MonoBehaviour component that matches the profile. Changes are type-specific — they apply to all instances of that MonoBehaviour type, not just one GameObject.

Q: Can I use this alongside existing custom editors?

Inspector Designer Pro registers its own custom editor for MonoBehaviour types that have profiles. If you already have a custom Editor class for a specific type, both may conflict. Remove or disable the custom Editor class to use Inspector Designer Pro for that type.

Q: Will this affect my builds?

No. The entire package is editor-only. It is excluded from all builds automatically through Assembly Definition platform settings. There is zero risk of build failures.

Q: How do I reset a field to default?

Use Ctrl+Z to undo recent changes. To fully reset a field, delete the profile asset from your project (it will be recreated fresh the next time you customize a field).

Q: Which field types can I use with which properties?

The UI automatically shows only compatible field types for each property. For example:

- int/float: Slider, Knob, ProgressBar, PercentageField, ButtonTabs, Dropdown
- bool: ToggleButton
- Enum: Dropdown, ButtonTabs
- string: Default only
- Texture2D/Sprite: ImagePreview
- AnimationCurve: CurveEditor
- Arrays/Lists: ReorderableList

Q: Can I export/share profiles between projects?

Yes. Profile assets are standard .asset files. Copy them between projects, making sure to also copy the .meta files.

Q: Does this work in Unity 2020?

We recommend Unity 2021.3 LTS or later for the best experience. Older versions may work but are not officially supported.

18. Support & Contact

If you encounter any issues, have feature requests, or need help:

- Email: ayushcodemate@gmail.com
- Leave a review on the Unity Asset Store — it helps other developers find this tool!
- Report bugs with a screenshot or screen recording if possible.

Thank you for using Inspector Designer Pro!

We hope it saves you hours of custom editor code and makes your inspectors beautiful.