🎩 Clockwork UI – Main Menu System v2.0

“Where control meets style.”
Clockwork UI is a controller-first main menu system. Built for Unity 6 and styled with a moody, mechanical elegance, it delivers cinematic-quality menus, smooth navigation, and fully modular UI components — all built for consoles and gamepads only.

🧭 Overview

Clockwork UI v2.0 is a **controller-exclusive**, fully modular Unity UI framework tailored for stylish, immersive main menu and settings systems. It is designed to be developer-friendly and visually polished, with all key elements automatically detected and managed via a powerful centralized manager.

Whether you're working on a Souls-like or a stylized AAA experience, Clockwork UI gives you a rich, responsive, and adaptable menu experience out of the box.

✨ Features

- 🎮 **Controller-Only Navigation** – Full support using Unity’s Input System (no mouse/keyboard).
- 🧠 **Smart Detection** – Panels and buttons are auto-detected and managed.
- 🧩 **Custom UI Elements** – Dropdowns, sliders, toggles, buttons – all themed and animation-ready.
- 🔥 **DOTween-Powered Transitions** – Flame selection effects, smooth fades, and transitions rely on DOTween.
- 📁 **Clean Project Structure** – Optimized folders for assets, animations, audio, and code.
- 🚀 **Unity 6 Ready** – Built for Unity 6000.0 and beyond.

📁 Folder Structure

```
Clockwork UI Main Menu
 ├── Animation        # DOTween-based anims + animator assets
 ├── Audio            # SFX for menu navigation and interaction
 ├── Fonts            # UI font assets
 ├── Images           # Icons, background images, button graphics
 ├── Scripts          # Full source for all menu systems and elements
```

🔧 **Note:** The Prefabs folder has been removed in v2.0 — all configuration is handled via drag-and-drop in the scene using the new detection system.

⚙️ Integration Steps

1. Import the Package

Install the package into your project (Unity 6 or newer recommended).

2. Scene Setup

- All this will be under The Canavas
- Add an empty GameObject named **Panels**.
- Place all your main menu panels (Main Menu, Settings, Credits, etc.) as children of Panels.
- Tag each with **UIPanel** (required).

3. Add the Core Script

- Drag the AdvancedUIMenuManager script onto a GameObject in your scene (recommended: an empty GameObject named UIManager).
- It will:
    - Automatically detect all UIPanel objects.
    - Auto-detect buttons inside each panel.

🎮 Panel & Navigation Logic

Feature	Description
Panel Detection	All child objects of Panels tagged with UIPanel are registered
Button Detection	Buttons under each panel are dynamically tracked and managed
Default Selection	You can assign default selection for each panel via Inspector
Flame Animation	Smoothly animated using **DOTween** — required for proper visual flow

---

🛠 Core Scripts & Usage

Script	Function
AdvancedUIMenuManager	The central manager that detects all panels/buttons and controls navigation
SettingsTabManager	Core logic for tabbed Settings menu (e.g., Gameplay, Graphics, Audio)
PanelSwitchManager	Handles switching between menu panels by calling public functions
AutoScrollToSelected	Keeps selected elements in view when inside scrollable menus
BringToFront	Forces UI elements (like dropdowns) to render in front of other components
ButtonSelectionActivator	Triggers a GameObject when its associated button is selected
CustomDropdown	Fully featured dropdown with animations and controller support
CustomSliderController	Interactive slider with feedback and value display

Script	Function
CustomToggle	On/off toggle switch with smooth animation
InputActionPerformer	Triggers events when certain input actions occur (e.g., “Back” button)
MenuHapticFeedback	Adds haptic feedback when navigating UI with a controller
OptionEventTrigger	Lets you trigger events from dropdown options

---

🧱 Settings Menu Tabs

Use **SettingsTabManager** to create and manage a category-based Settings UI:

- Each tab (e.g., Audio, Video, Gameplay) has its own content area.
- Seamless navigation between tabs using the controller.
- Works automatically with CustomSlider, CustomDropdown, and CustomToggle.

🎨 Visual Design

- **Color Theme:** Dark steely greys, soft glows, Red
- **Fonts:** Gothic serif and minimal sans-serif font pairings
- **Motion:** Every interaction is backed by subtle DOTween transitions and fades
- **Feedback:** Includes haptic feedback and SFX for tactile responsiveness

---

🔗 Dependencies

Dependency	Used For
Unity 6.0+	Minimum supported version
TextMesh Pro	All text rendering

Dependency	Used For
New Input System	Controller input handling (REQUIRED)
DOTween Pro	All UI transitions, selection flame, animations (REQUIRED)

🧠 Important Notes

- ⚠️ **Do Not Modify Core Scripts**
  They are self-contained and connected through detection logic.
- 🔧 **Customize in Inspector Only**
  Modify button behavior, audio clips, UI colors, and animation speeds via the Inspector.
- 🕹️ **Controller-Only Workflow**
  Mouse or keyboard interaction is not supported in this package.

🤖 AI Asset Use

Some visual assets (e.g., icons, textures, and fonts) have been created or enhanced using AI tools. All assets are 100% cleared for **free commercial use**.

📞 Support & Contact

Have questions or feedback?

📧 **Email:** [SoJoG.DEV@hotmail.com]
💬 Whether it's a bug report, feature request, or integration help — I'm here to assist!

Clockwork UI – v2.0 was built to power games that value atmosphere, control, and polish. Take your menu from placeholder to *playable art*.