

Back Button Manager

Overview

The **Back Button Manager** package enables you to manage back-button behavior directly from the Unity Editor without modifying your code. It also provides methods for runtime adjustments through scripting. It offers a flexible and adaptable structure suitable for various scenarios. With this package, you can easily configure actions triggered by the Android physical back button or the desktop Escape key.

Package Components

BackButtonController

- Captures back-button inputs, manages handlers, and forwards events to registered handlers.

BackButtonActionHandler

- Defines actions to perform when the back button is pressed.

ApplicationPauseControl

- Automatically pauses the game when the application is minimized or loses focus on mobile devices.
- **Note:** Attach this script to a GameObject that already contains a `BackButtonActionHandler` component.
- Includes editor-only pause simulation for convenient testing.

BackButtonUIScanner

- Scans all build-included scenes to display UI elements with the `BackButtonActionHandler` script and their configurations.
- Provides an overview of back-button actions used across your project.

Action Types

ButtonClick

Triggers a specified UI button action upon pressing the back button.

- **Tip:** If `autoRegisterOnStart` is set to `true`, the handler automatically registers when the `GameObject` becomes active.
- **Tip:** Use `targetActionHandler` to activate another handler after completing an action (e.g., toggling pause/resume).

Trigger

Activates another handler through a predefined `triggerTargetHandler` when pressed.

Remove

Removes handlers based on the chosen removal strategy.

- **Tip:** Using `removeHandlerType`, you can remove only the last registered handler, clear all handlers, or remove a specific handler.
 - **RemoveHandlerOnButtonClick:** Removes the most recent handler when the button is clicked. Useful for scenarios like closing a window where both OK and Cancel buttons close the window, but the back button specifically triggers the Cancel action. Clicking OK would remove the handler associated with Cancel.
 - **ClearAllHandlersOnButtonClick:** Clears all handlers when the button is clicked.
 - **RemoveHandlerOnStart:** Removes the next handler in the stack when the `GameObject` with this script becomes active.
 - **ClearAllHandlersOnStart:** Clears all handlers when the `GameObject` with this script becomes active.
 - **RemoveTargetHandler:** Removes the handler associated with a specified target object when the button is clicked.

Quit

Exits the application upon pressing the back button.

- **Tip:** Use `clearAllHandlersOnStart` in main menu-like scenes to clear all existing handlers upon loading.

Wait

Temporarily disables back-button functionality.

- **Tip:** Recommended for scenarios like modal popups.

BackButtonViewer

Provides an editor window for visually inspecting registered handlers and the quit/wait states.

- Open this viewer through the "Open Back Button Viewer" button on the GameObject with the `BackButtonController` component.

Installation

Add the Prefab to Your Scene

- Include the "Back Button Manager" prefab provided in the package into your initial scene. This prefab automatically includes the `BackButtonController` component.

Setting up Action Handlers

- Attach the `BackButtonActionHandler` component to UI elements (e.g., buttons or panels) where back-button functionality is desired.
- Configure the relevant settings via the Inspector.
- Refer to the included demo scene for detailed usage examples. You can also explore the provided demo scene and watch the tutorial video for additional guidance.

Usage

BackButtonActionHandler Configuration

- **ButtonClick:** Triggers specific UI button actions when the back button is pressed.

- Automatically registers when `autoRegisterOnStart` is `true`.
 - Activate another handler after action completion using `targetActionHandler`.
- **Trigger:** Activates another handler through `triggerTargetHandler` upon button press.
- **Remove:** Removes handlers based on the defined strategy (`removeHandlerType`).
 - See detailed descriptions under the Action Types section.
- **Quit:** Terminates the application upon pressing the back button.
 - Clear existing handlers on scene load with `clearAllHandlersOnStart` as needed.
- **Wait:** Temporarily disables the back button, suitable for modal dialogs.

BackButtonController Usage

- **Registration:** Handlers are managed using a stack structure.
 - Register handlers using the `RegisterHandler` method.
- **Removal:** Remove specific handlers or all handlers using the `UnregisterHandler` methods.
- **Mode Configuration:** Activate or deactivate quit and wait modes using `SetQuit` and `SetWait` methods.

ApplicationPauseControl Usage

- Attach this script to GameObjects that have a `BackButtonActionHandler` component.
- Automatically triggers the pause action when the application loses focus or is minimized (on mobile devices).
- In the Unity Editor, you can simulate pause functionality by pressing the configured shortcut key (default: `P`).

BackButtonUIScanner Usage

- Use this tool to scan all scenes included in the build.
- Provides an overview of UI elements with assigned `BackButtonActionHandler` scripts and their configurations.
- Helpful for ensuring consistent back-button behavior across your project.

Customization and Integration

Back Button Manager allows extensive customization of back-button behaviors both via the Unity Editor and through runtime scripting methods. Combine different action types and removal strategies to tailor interactions perfectly suited to your UI flow. Additionally, use the `BackButtonViewer` and `BackButtonUIScanner` tools to visually test, inspect, and validate system states directly within the Unity Editor. Refer to the demo scene and tutorial video for practical examples and detailed explanations.