UI Animation Components – Usage Guide

This document describes how to apply and configure the UI animation components used in the project. The components are designed to be composable, predictable, and configured primarily through the Unity Inspector.

General Principles

Each component is responsible for a single aspect of animation control. Components are combined on a single GameObject and connected via UnityEvent to form animation flows. Animator Controllers are kept simple and reusable.

AnimatorDelayActivator

Purpose: Controls when the Animator becomes active. Prevents visual glitches and allows delayed or randomized animation start.

- Add the component to a GameObject with an Animator.
- Choose fixed delay or random delay using the inspector toggle.
- Optional: enable deterministic mode for stable random values.
- Use the On Started UnityEvent to trigger the next step in the animation flow.

AnimatorAutoSwitchOnComplete

Purpose: Switches the Animator Controller after a specific state has completed. Used for sequential animation flows such as fade-in followed by idle or scale animation.

- Set the Animator reference.
- Assign the next Animator Controller.
- Specify the state name to wait for completion.
- Use the On Switched UnityEvent to trigger state playback.

AnimatorStateRouter

Purpose: Explicitly starts a specific Animator state. Useful when Entry does not automatically transition to a playable state.

- Assign the target state name in the inspector.
- Disable Play On Enable if state should be started via UnityEvent.
- Connect PlayTarget() to a UnityEvent from another component.

AnimatorRandomStart

Purpose: Desynchronizes repeated UI animations by applying random delay, speed, and normalized start time.

- Enable or disable random delay.
- Optionally randomize animation speed.
- Enable normalized time randomization for phase offset.

Example Animation Flow

A typical UI animation flow consists of the following steps:

- AnimatorDelayActivator waits before enabling Animator.
- UIShow animation plays.
- AnimatorAutoSwitchOnComplete switches Animator Controller.

- AnimatorStateRouter starts the required state.

All configuration is done through the Inspector, allowing different behaviors per UI element while reusing the same Animator Controllers.