Design Guidelines

Guidelines for integrating CommonUI and using it in your project.

CommonUI is an extensive addition to UMG with a large variety of systems and tools to understand. This page contains a series of guidelines and frequently asked questions to help you learn the best practices for implementing it.

Is CommonUI right for your project?

While CommonUI has many beneficial features, it is built with two main use cases :

- Complex multi-layered interfaces.
- Cross-platform support.

If you do not anticipate your project using either of these use cases, then CommonUI might not be necessary. Beyond these use cases, we recommend that you consider what benefits are driving you to using CommonUI, and if those benefits are worth the cost of learning a new paradigm for UI creation and interaction.

As an example, when designing a PC-only real-time strategy (RTS) game, CommonUI might not be applicable, as this type of game generally uses a single-layered UI that doesn't need

complex cardinal navigation, therefore it is unlikely to benefit from using CommonUI's Input Routing or Bound Action systems.

Additionally, CommonUI is not recommended for use with widgets placed using WidgetComponents. Since CommonUI relies on cursor-focus navigation, activation order, and paint order/layer ID, CommonUI can still handle your 2D game HUD, but it does not work with widgets placed in your game's world.

Migrating to CommonUI

If you have an existing UI, you may be considering migrating to CommonUI. When making this decision, consider if your UI is nearly complete and if there are future UI development plans that can benefit from using CommonUI.

Additionally, consider if it is possible to create new widgets with CommonUI while interacting with your old UI, or if an entire UI refresh would be needed to migrate to CommonUI.

Should I use CommonActivatableWidget or CommonUserWidget?

Not every CommonUI widget should be an Activatable Widget. A widget should only be an activatable widget if it needs to affect Input Routing on an on or off basis.

If the widget you are creating only needs to interact with CommonUI's Input Routing system to handle input by itself, consider creating a CommonUserWidget or a regular User Widget. CommonUserWidgets form the basis of many of CommonUI's classes, including `UCommonButtonBase` and `UCommonTabListWidgetBase`. Tooltips are a good example of a case where Activatable Widgets can be counterproductive since tooltips tend to quickly appear and disappear, don't need to forward input to child widgets, and don't need to seize input from the rest of your UI.

If the widget you are creating has multiple interactable children or needs to block input handling from the rest of your UI, then using an Activatable Widget is preferable. Pop-up windows and modal menus are good examples of this kind of behavior.

How do I handle simultaneous keyboard and mouse navigation?

CommonUI easily supports gamepad navigation as well as conventional mouse and keyboard navigation. A common pitfall is wanting to support **simultaneous** keyboard and mouse navigation. Common UI does **not** support navigating with your mouse while simultaneously navigating with the keyboard as if it were a gamepad. CommonUI.

For this particular combination of navigation inputs, consider the following: if your mouse can freely navigate and have its own hover/focus separate from the keyboard, what happens when your keyboard navigates to an element not currently hovered/focused by the mouse? Your UI would appear in a state where a single player is hovering two separate elements. This state would be visually confusing and hard to support across all UIs in a game.

While CommonUI does not directly support simultaneous keyboard and mouse navigation, there is nothing preventing you from creating specialized widgets to behave correctly with keyboard navigation. Likewise, there are no design issues when _toggling _between using mouse navigation and using the keyboard as a gamepad for navigation, because toggling maintains that one widget has focus at all times.