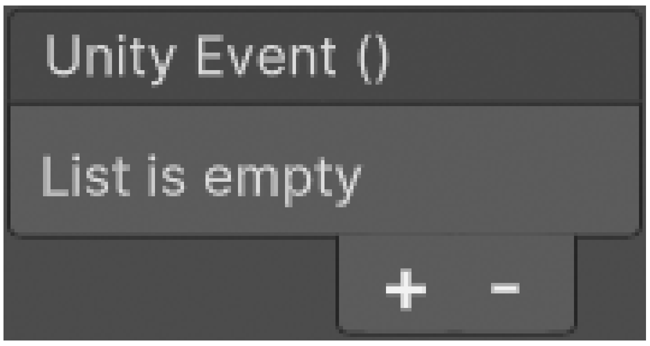
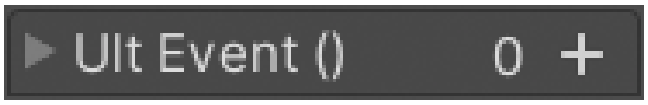
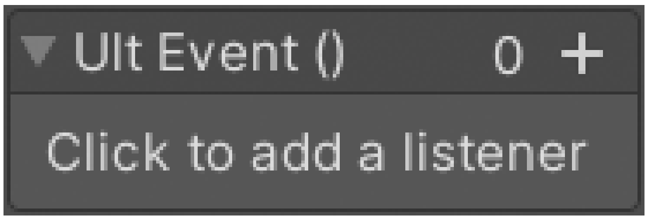
`UnityEvents` are a system built into Unity which allows users to easily setup and configure persistent event callbacks via the Inspector.

`UltEvents` serve the same purpose, but with an improved user interface, better features, and fewer restrictions.







Serializable Event System Feature Comparison	**UnityEvents**	**UltEvents**

Features

Persistent listeners (serialized)		
Dynamic listeners (non-serialized)		
Call methods with 0 or 1 parameter		
Call methods with any number of parameters		
Parameter types: bool, int, float, string, UnityEngine.Object		
Parameter types: Enum (regular and flags), Vector (2, 3, 4), Quaternion, Rect, Color, Color32		
Call methods with non-void return types		
Use returned values as parameters in subsequent calls		
Call public methods		
Call non-public methods		
Call static methods		
Get and Set fields directly		
Source code included		
Disable specific listeners entirely or just in Edit Mode		[1]

User Interface

Searchable menu for method selection		

Serializable Event System Feature Comparison	UnityEvents	UltEvents
Compact and collapsible GUI		
Parameterless functions only take a single line		
Function parameter names are displayed		
Customizable display options		
Keyboard shortcuts (Copy, Paste, Add, Delete)		
Context menu commands (Invoke, Clear, Copy, Paste)		
Select a specific component when there are multiple of the same type		
Button to quickly find a similarly named method if the target is missing		
Displays dynamic (non-serialized) listeners in the GUI		

[1] `UnityEvent`s have a dropdown menu for each persistent listener to select when it should be executed. The default is `Runtime Only` which will skip that listener if the event is invoked in Edit Mode in the Unity Editor. `Off` can also be used to disable that listener entirely. In the development of `UltEvent (/ultevents/api/UltEvents/UltEvent)`s this feature was deemed to not be particularly useful so it was skipped to avoid it's implementation/maintenance/performance cost.

Performance

- `UltEvent (/ultevents/api/UltEvents/UltEvent)`s and `UnityEvent`s both have significantly worse performance than regular C# Delegates (https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/delegates/) like `System.Action`. If you only want an event that can be used in code without wanting to set it up in the Inspector or save it as part of a scene/prefab, then you should simply use regular delegates.
- When calling methods with no parameters, `UltEvent (/ultevents/api/UltEvents/UltEvent)`s are slightly faster than `UnityEvent`s.
- When calling methods with one parameter, `UnityEvent`s are slightly faster than `UltEvent (/ultevents/api/UltEvents/UltEvent)`s.
- Methods with more than one parameter simply cannot be called by `UnityEvent`s.
- The package includes some simple performance testing scripts in the `UltEvents.Benchmarks` namespace.

Documentation	Download	Contact
Documentation (/ultevents/docs)	**Itch.io (Recommended) (https://kybernetik.itch.io/ultevents)**	**Unity Forum (https://forum.unity.com/thı**
Change Log (/ultevents/docs/changes)	**Asset Store (https://assetstore.unity.com/packages/tools/gui/ultevents-111307?aid=1100l8ah5)**	**Email (mailto:mail@kyberne-subject=UltEvents)**