Dynamic Listeners

You can easily register delegates to an event using the `+=` operator like a standard C# delegate:

```
_MyEvent += SomeMethod;// Existing Method.
_MyEvent += () => ...;// Lambda Expression.
```

Normally this will register the delegate as a dynamic listener (non-serialized), however if you are currently in Edit Mode (in the Unity Editor but not in Play Mode) it will register as a persistent listener instead.

To specifically add a dynamic listener, you can either access the `dynamicCalls` field directly or use the `static AddDynamicCall` method which will automatically null-check the event:

```
_MyEvent.dynamicCalls += SomeMethod;
UltEvent.AddDynamicCall(ref _MyEvent, SomeMethod);
```

Configuring Persistent Listeners Using Code

To specifically add a persistent listener, you can either create a `new PersistentCall` `(/ultevents/api/UltEvents/PersistentCall)` and add it to the `UltEventBase.PersistentCalls` list or you can call `UltEventBase.AddPersistentCall` which returns a `PersistentCall` `(/ultevents/api/UltEvents/PersistentCall)`.

```
// Make the call yourself:
var call = new PersistentCall((Action<float>)SomeMethod);
_MyEvent.PersistentCalls.Add(call);

// Or let it be created for you:
var call = _MyEvent.AddPersistentCall((Action<float>)SomeMethod);
```

You can then configure the call using `SetArguments` or by accessing the `PersistentArguments` array directly.

```
call.SetArguments(2);
call.PersistentArguments[0].Float = 2;
```

All of these `+` operators and `Add` methods have corresponding `-` operators and `Remove` methods.

Documentation

Documentation (/ultevents/docs)

Change Log (/ultevents/docs/changes)

Download

Itch.io (Recommended) (https://kybernetik.itch.io/ultevents)

Asset Store (https://assetstore.unity.com/packages/tools/gui/ultevents-111307?aid=1100l8ah5)

Contact

Unity Forum (https://forum.unity.com/thi

Email (mailto:mail@kyberne -subject=UltEvents)