

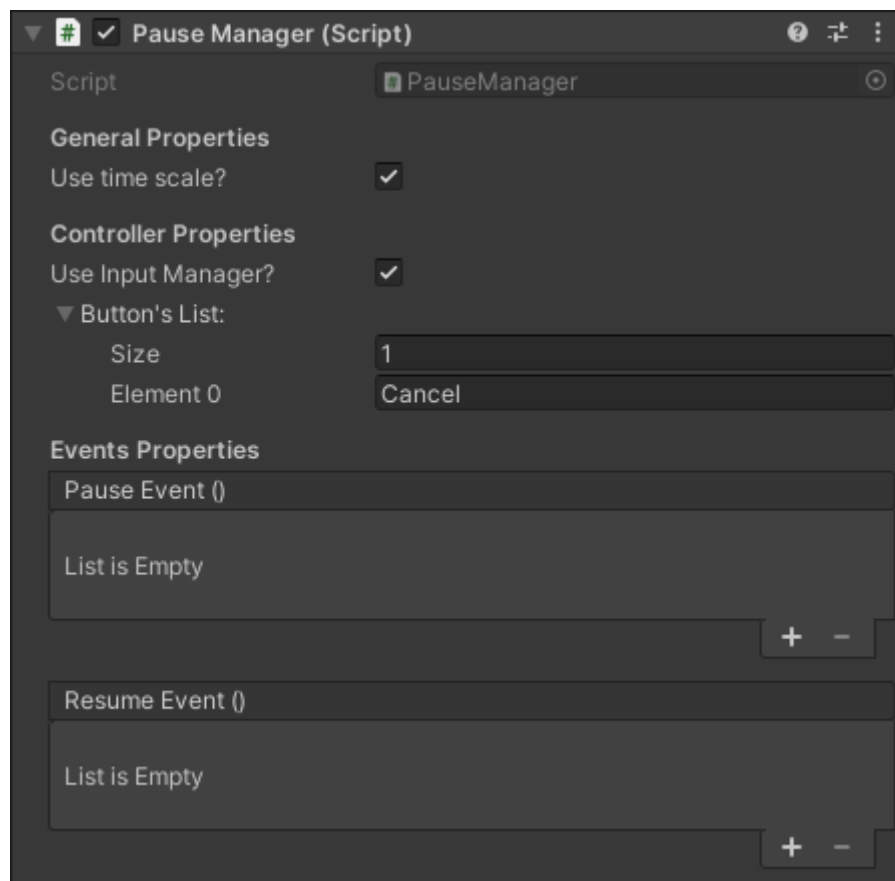
PAUSE MANAGER

Overview.....	3
Properties.....	4
General Properties.....	4
Use time scale?.....	4
Pause on blur?.....	4
Controller Properties.....	5
Input Manager.....	5
Input System.....	6
Use Player Input?.....	7
Use reference?.....	10
Pause Action.....	12
Rewired.....	14
PlayerPrefs.....	16
Pause Keys.....	17
Steam Overlay Properties.....	18
Events Properties.....	19
Pause Event Handler.....	20
Examples.....	20

Overview

Thanks for purchasing the **Pause Manager** package. It's a simple and free package for managing pause/resume in your app, it's easy to use and fully integrated with **Unity's Input System** and **Rewired**.

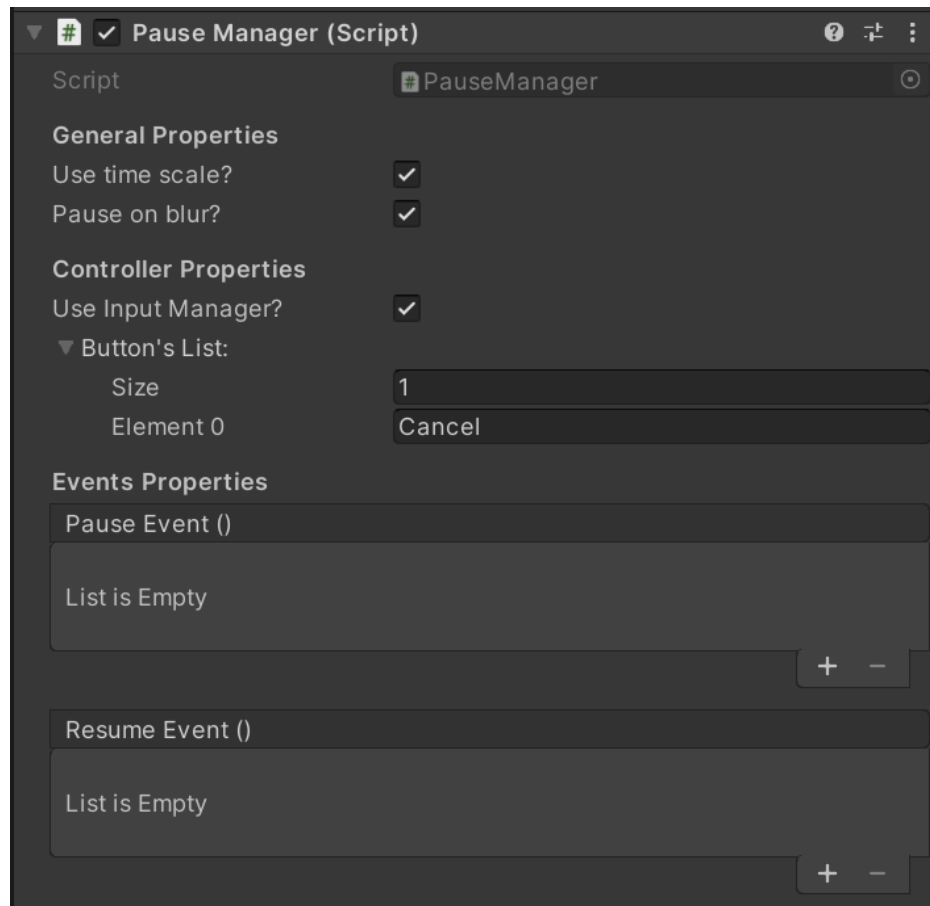
Simply drag-n-drop the **Pause Manager** prefab into your scene and you'll see the default options on Inspector.



Properties

General Properties

Under **General Properties**, there is a **Use time scale?** checkbox and a **Pause on blur?** checkbox (see image below).



Use time scale?

With this property checked, when you trigger **pause** events, the value of **Time.timeScale** will be 0 and, when you trigger **resume** events, the value of **Time.timeScale** will be 1.

Pause on blur?

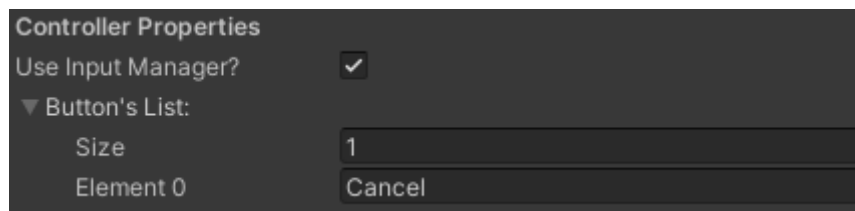
With this property checked, when the game window loses focus (blur) and the game is not paused, it will automatically pause.

Controller Properties

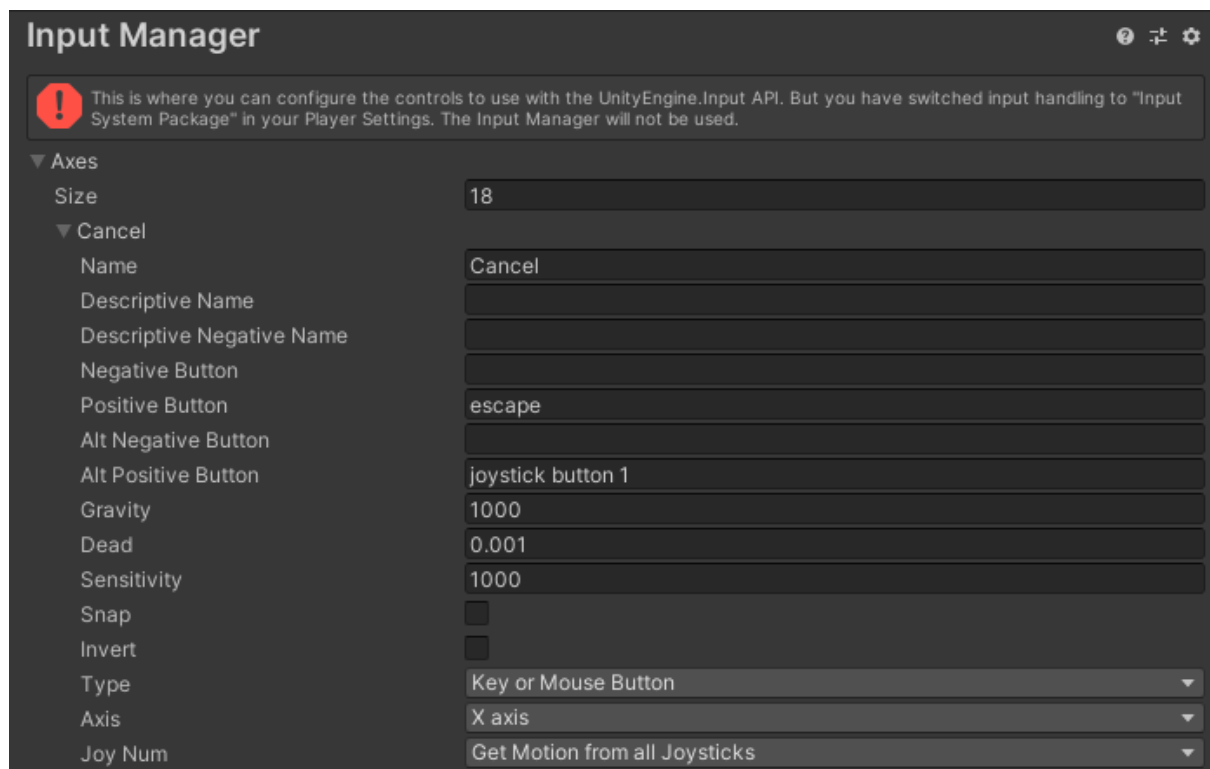
Input Manager

The **Button's List** specifies all buttons that could trigger the pause/resume events. Some example values could be **Player 1 Pause**, **Player 2 Pause** etc, where you configure the keyboard's key, Playstation controller's button, Xbox controller's button, etc.

By default, there's only one button with the **Cancel** value.

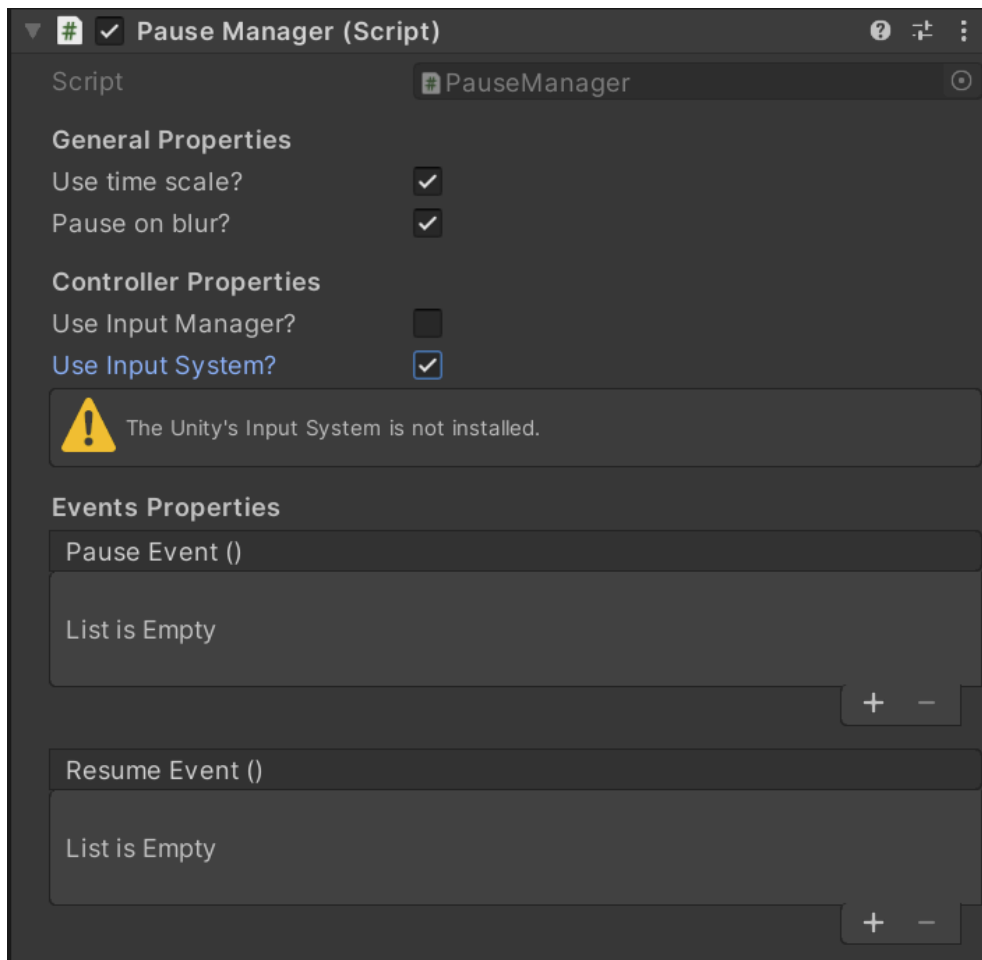


The **Cancel** button can be found in **Edit > Project Settings > Input Manager**.



Input System

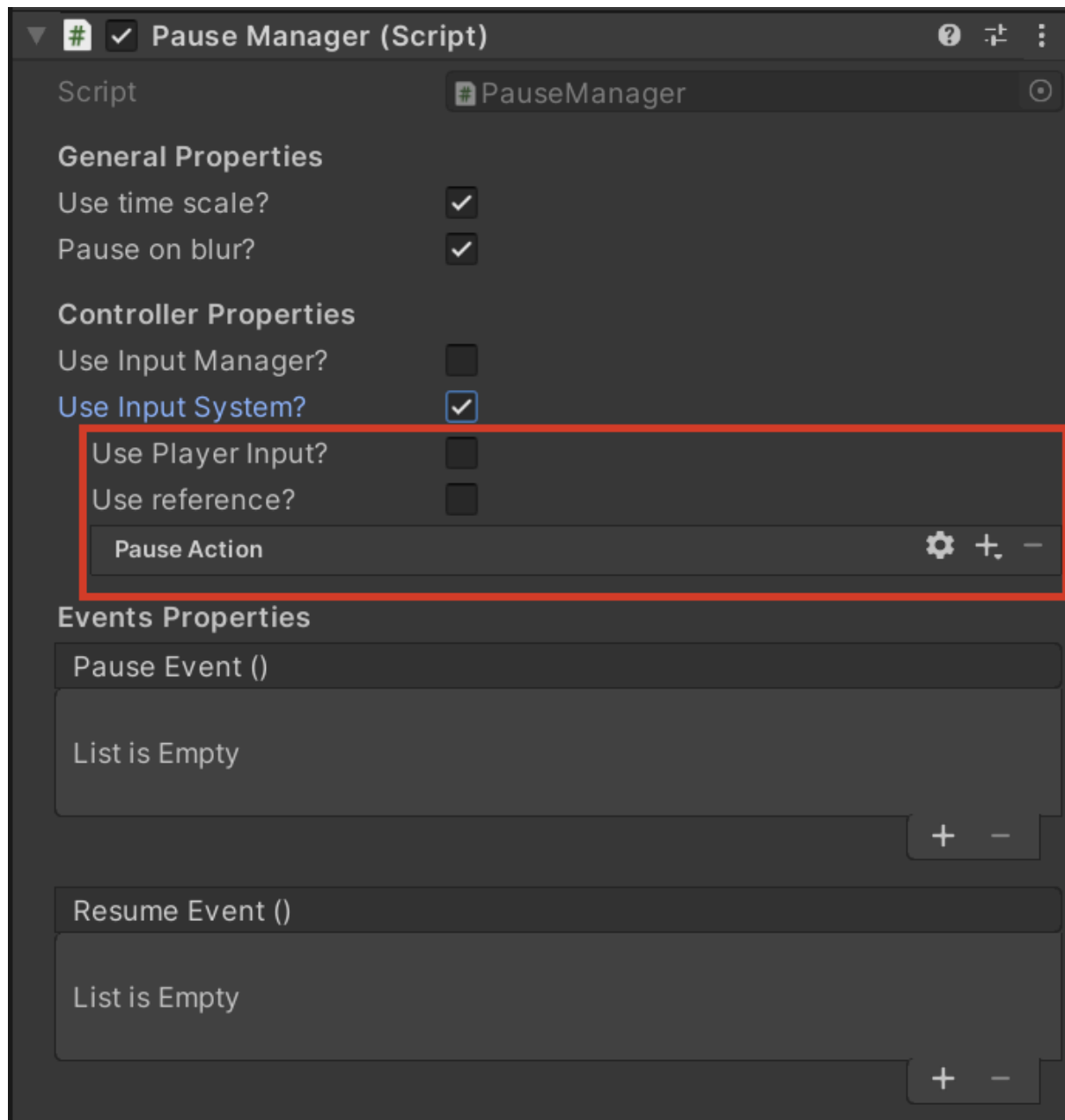
The default controller properties allows you to use Unity's current Input Manager, but if you want to use **Input System**, you can uncheck the **Use Input Manager?** checkbox which gives you these options below



Notice that if you do not have **Input System** installed, a warning message will appear. If you choose to install **Input System**, the minimum version required for this package to work is **0.2.10-preview**, but it is recommended that you install the latest version available.

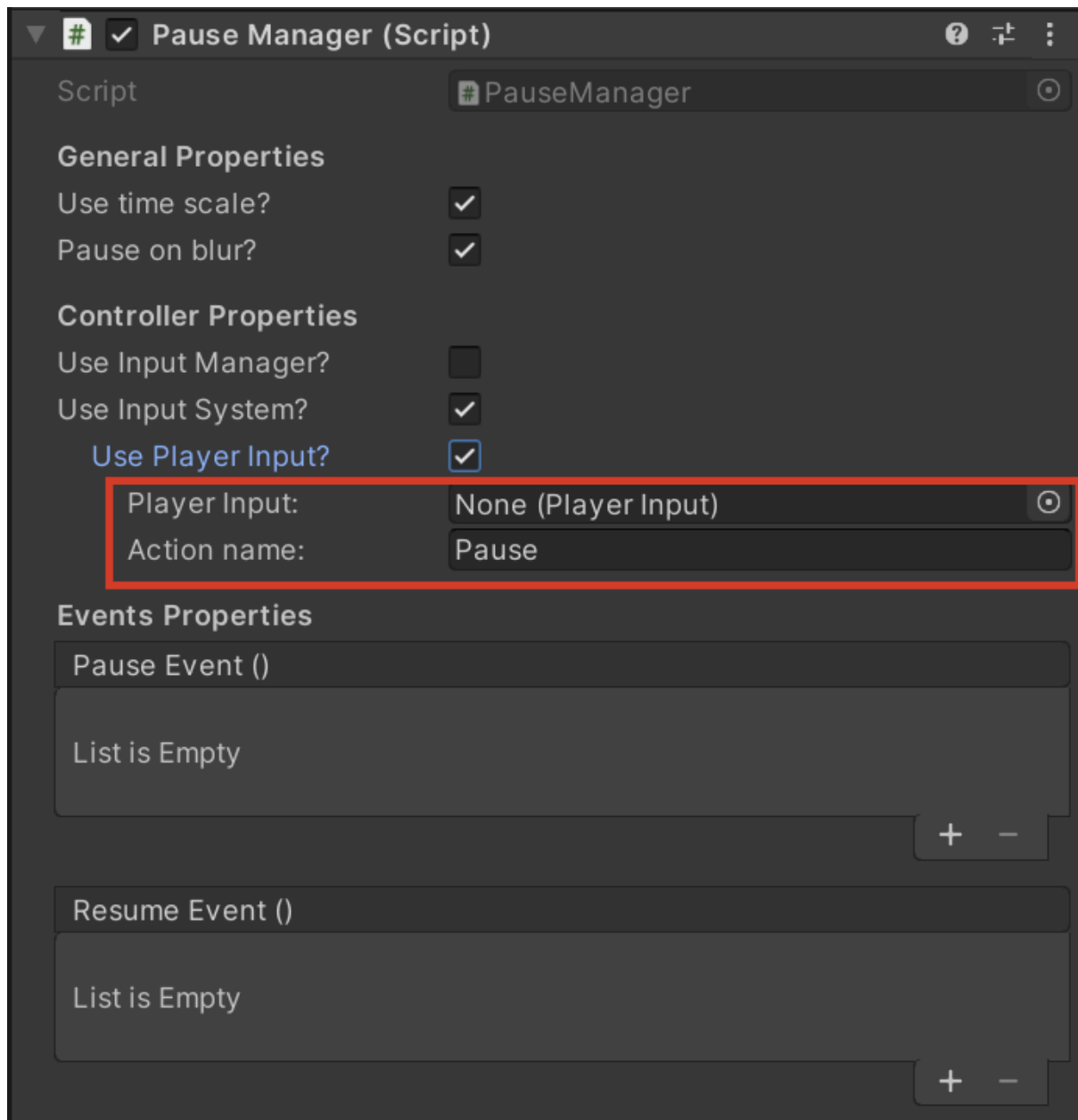
If you could not install the **Input System** package in your project, you don't need to continue these steps. You can still use the **Input Manager**.

After installing the **Input System**, you'll see the information available below.

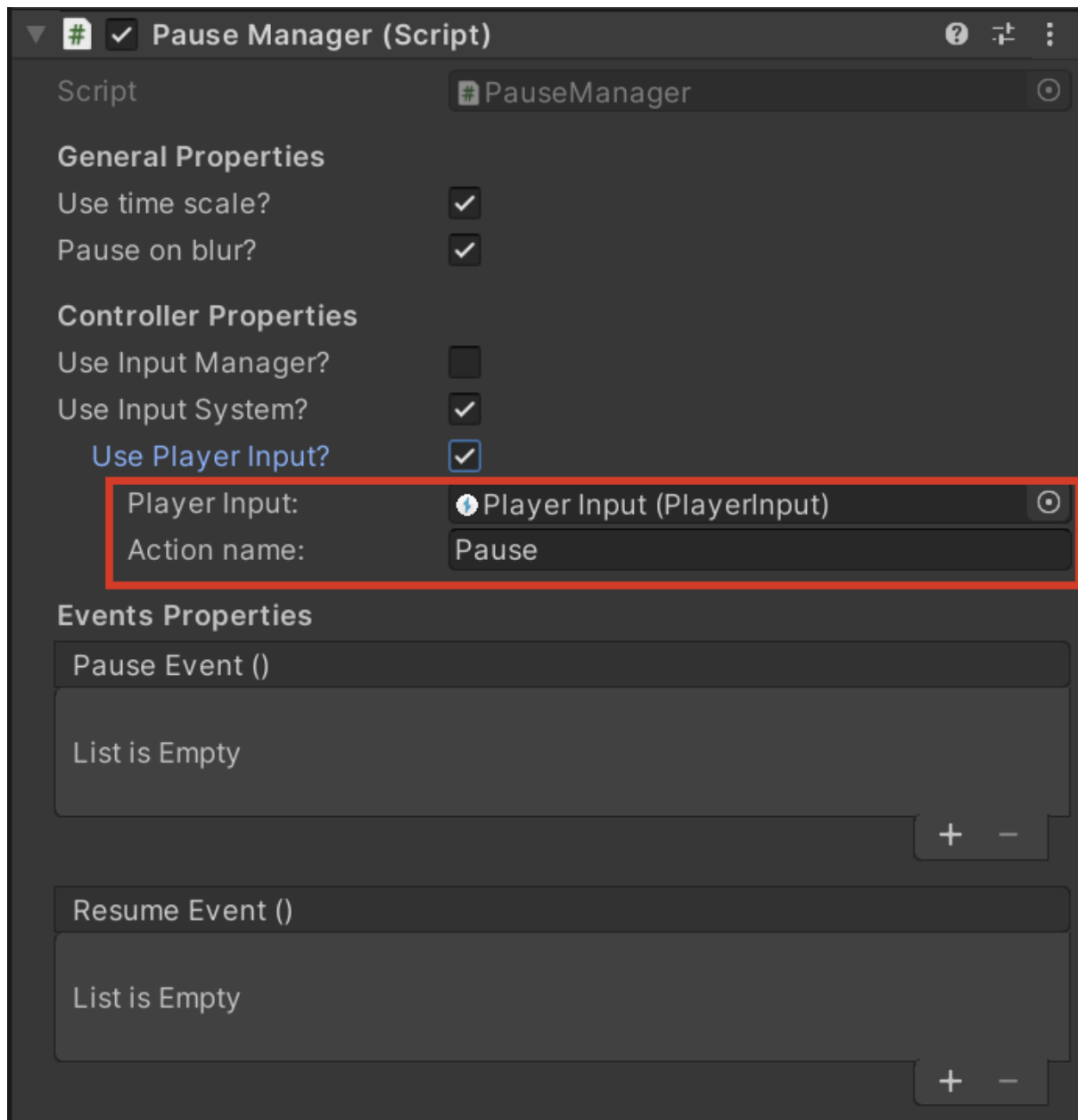


Use Player Input?

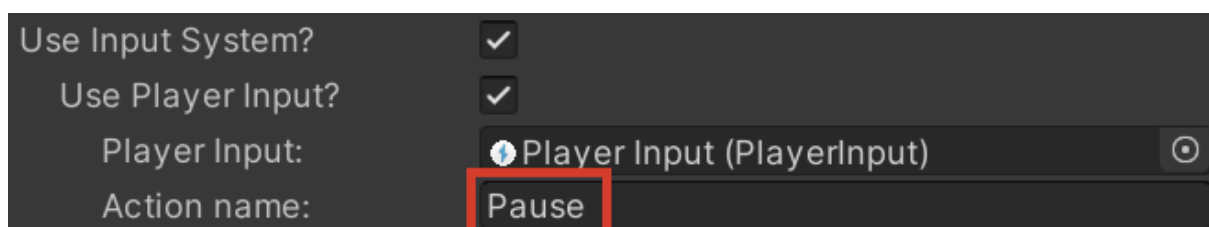
When you check the **Use Player Input?** field, you'll see **Player Input** and **Action name** fields.

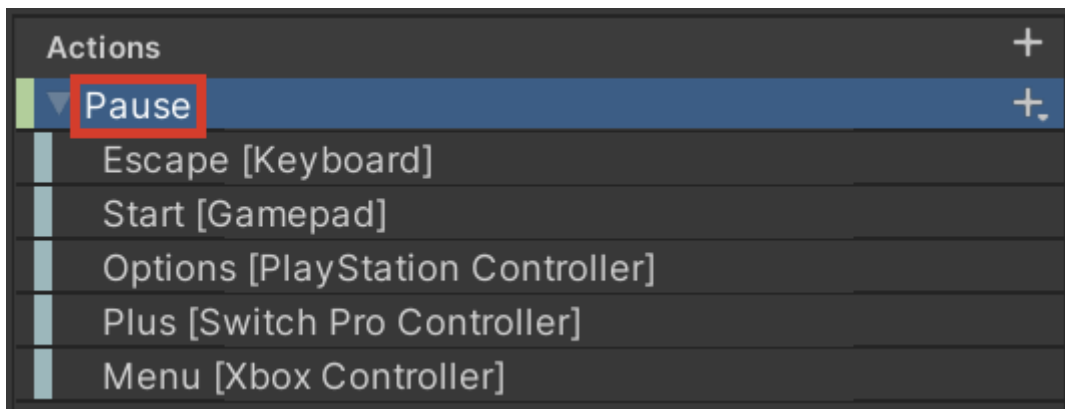


Just drag a GameObject with a **Player Input** script attached and input all sets.



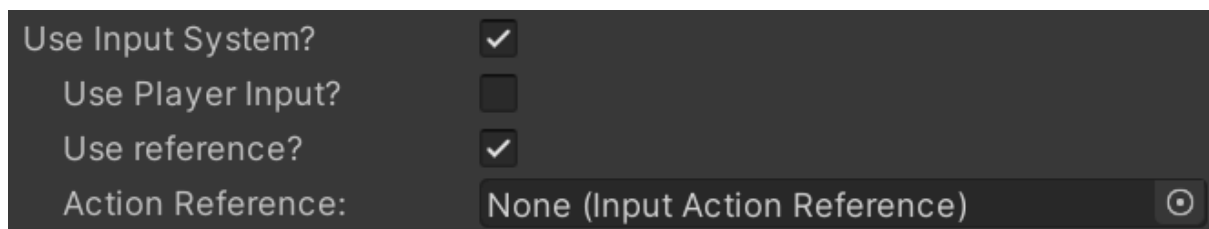
The **Action name** field value must match an action on the **Input Action Asset** file (see images below).





Use reference?

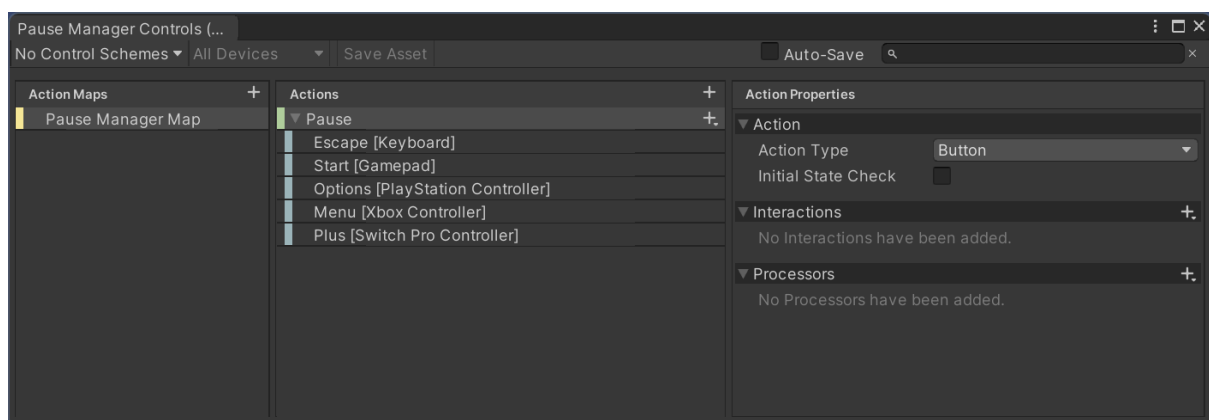
When you check the **Use reference?** field, you'll see the **Action Reference** field.



If you want to use an **Input Action Reference** from an **Input Action Asset**, just drag-n-drop from that asset in the **Action Reference** field.

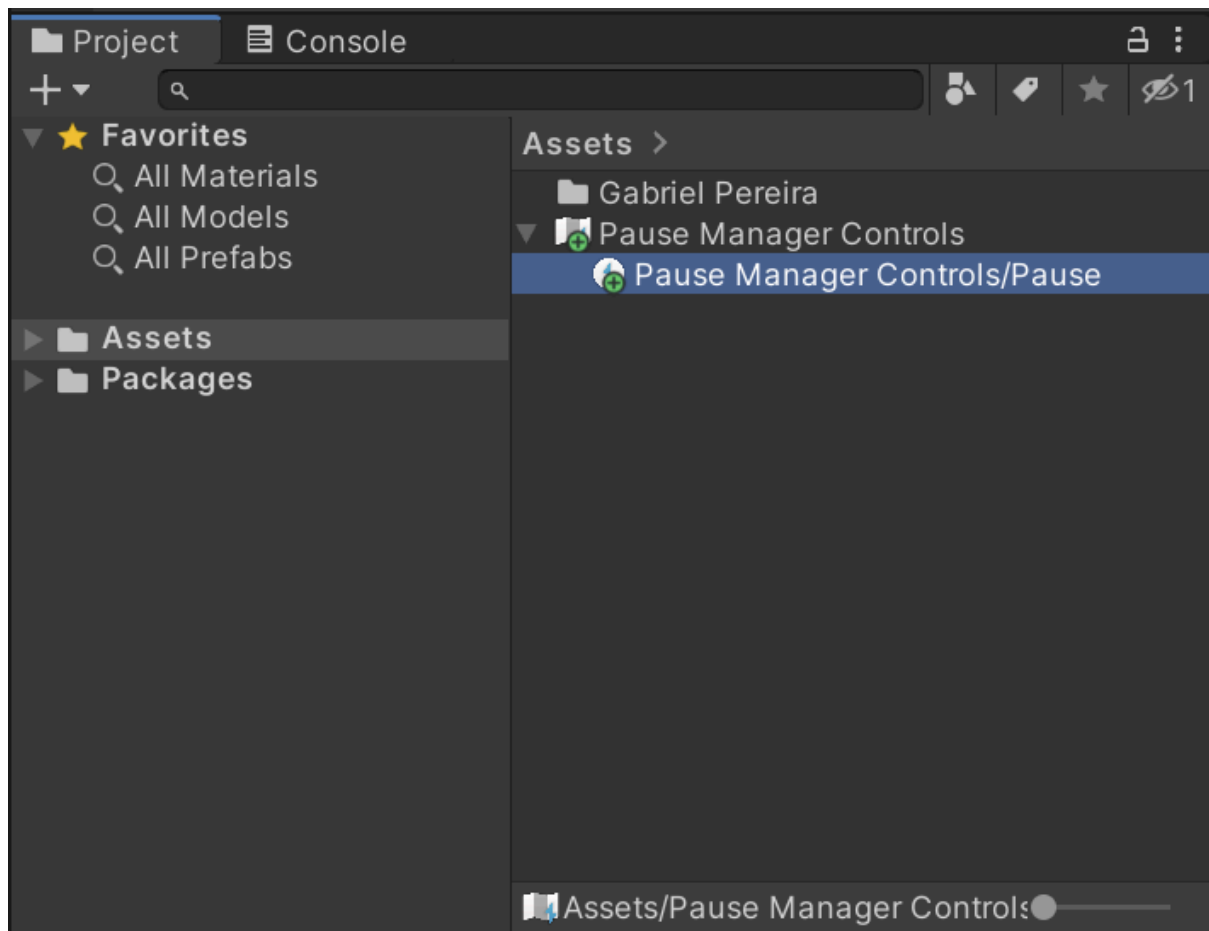
To achieve this, just follow these steps:

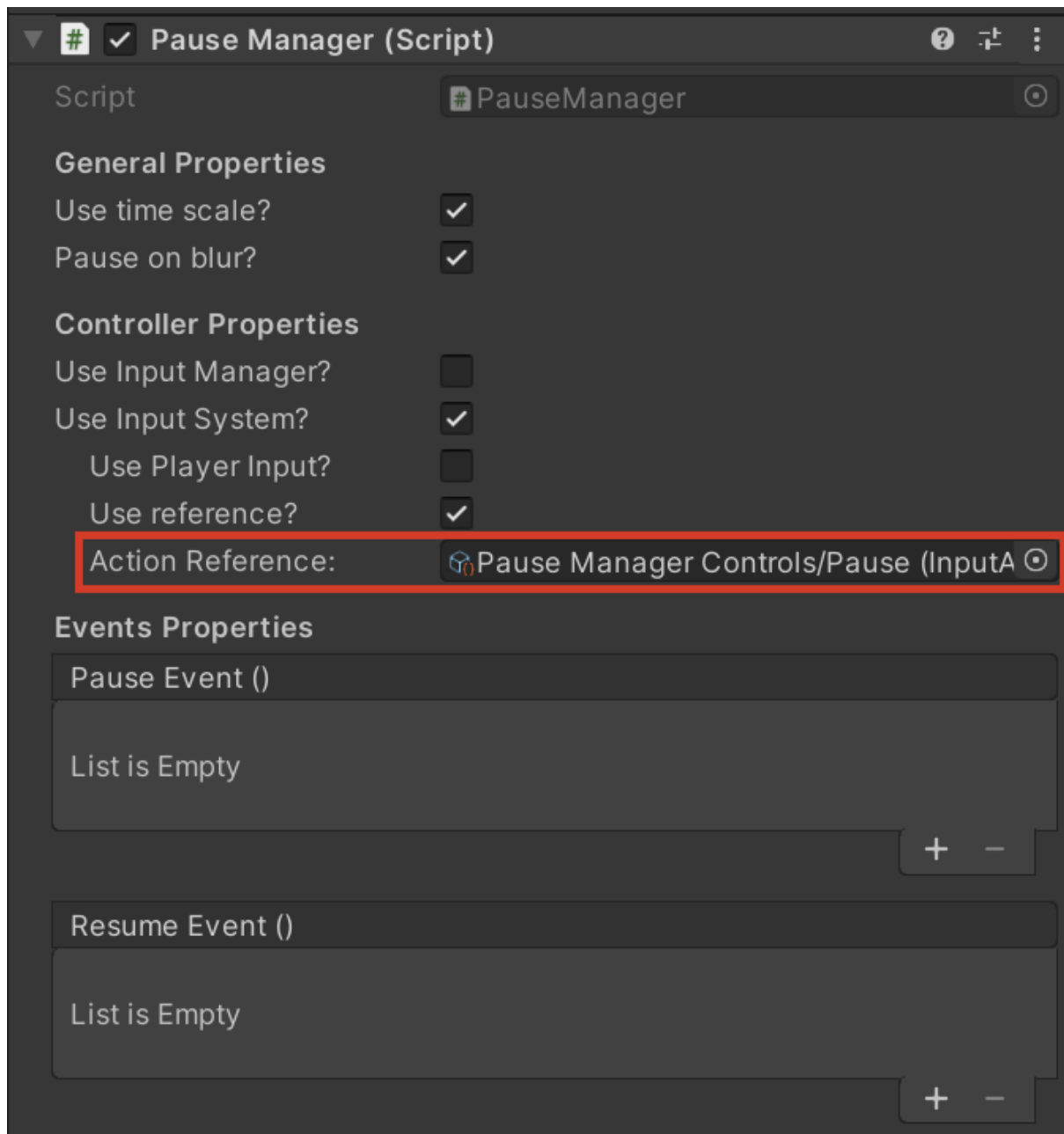
1. Create an **Input Action Asset**
2. Add an **Action Map** and its **Actions**
3. Map all the necessary buttons (see image below)



4. Click **Save Asset** (or check **Auto-Save**)

5. Drag the **Action Reference** created into the **Action Reference** field

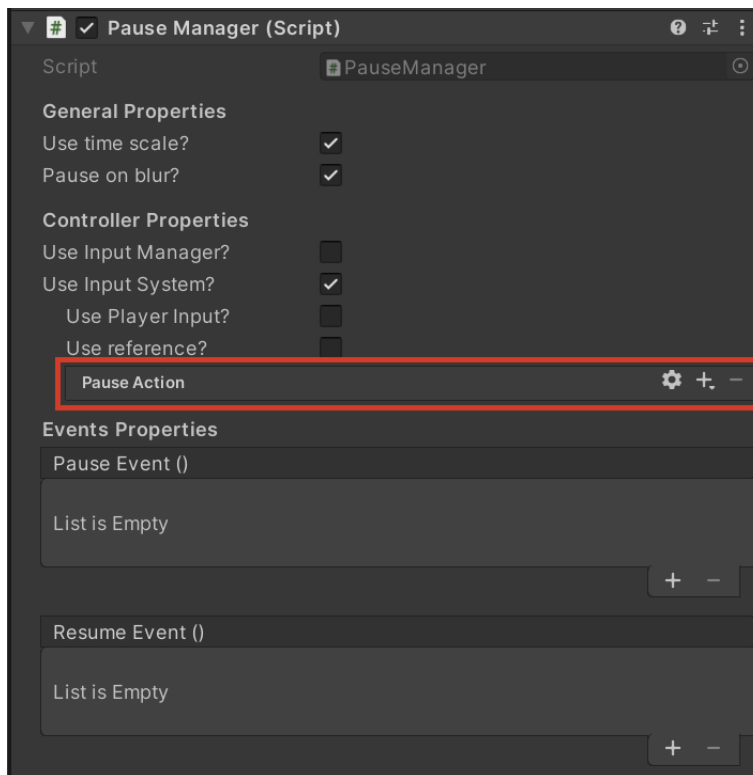




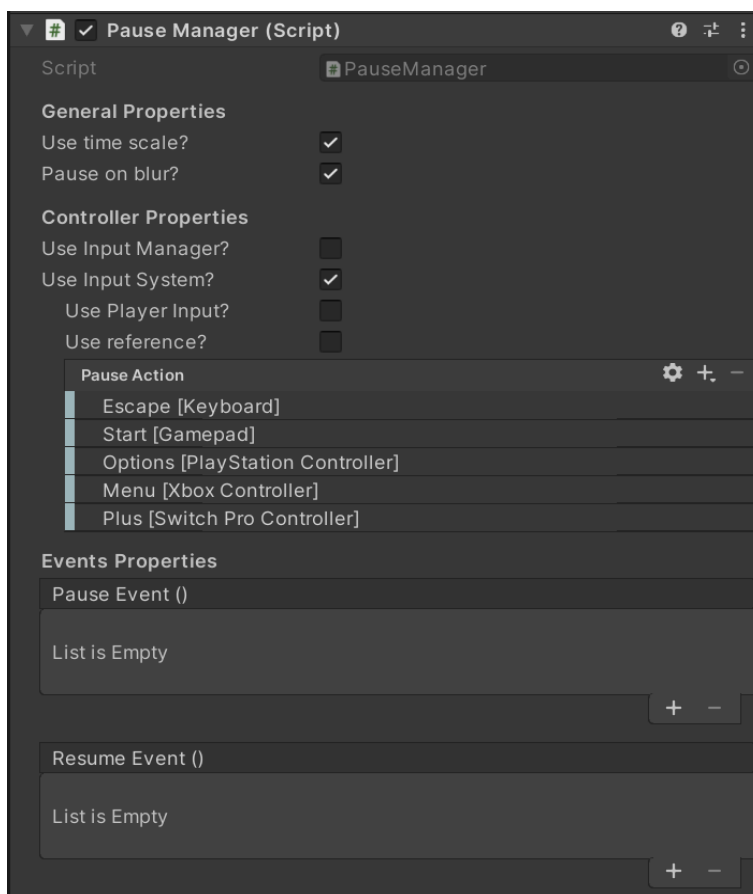
With this, **Pause Manager** will use the reference to trigger pause/resume events.

Pause Action

If none of the options above suits your need, there's the **Pause Action** field available below.



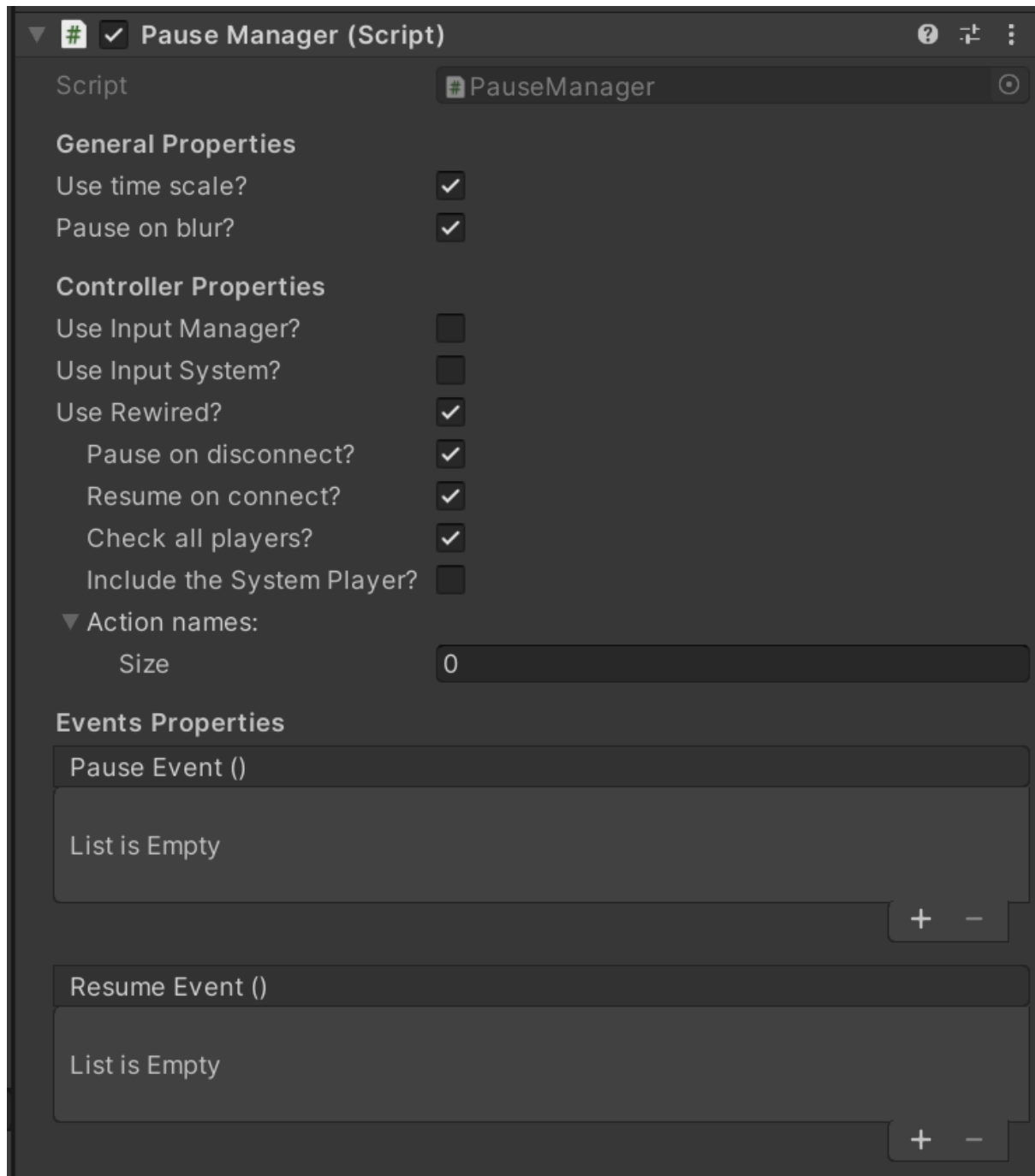
You can add some options to suit your game needs, for example:



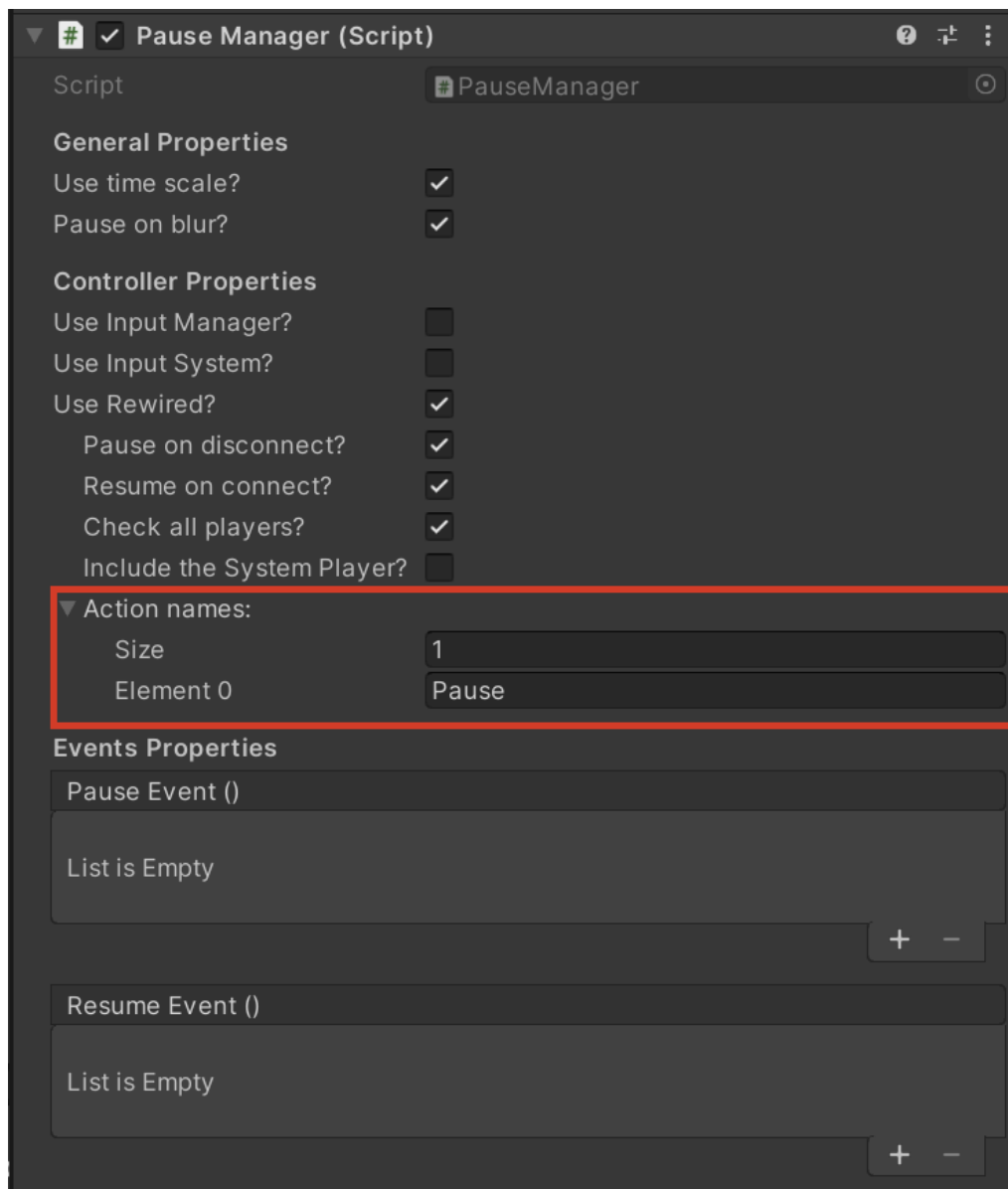
Now, if you press one of the buttons listed in the **Pause Action**, it will trigger the pause/resume events.

Rewired

The [Rewired](#) asset, made by [Guavaman Enterprises](#), is also supported by **Pause Manager**. To use Rewired, just check the **Use Rewired?** checkbox.



Now, just inform all the **action** names responsible for the pause/resume events.



▼ # ✓ Pause Manager (Script) ? ⚙ ⋮

Script # PauseManager

General Properties

Use time scale? ☒

Pause on blur? ☒

Controller Properties

Use Input Manager? ☐

Use Input System? ☐

Use Rewired? ☒

 Pause on disconnect? ☒

 Resume on connect? ☒

 Check all players? ☒

 Include the System Player? ☐

▼ Action names:

Size	1
Element 0	Pause

Events Properties

Pause Event ()

List is Empty

+ -

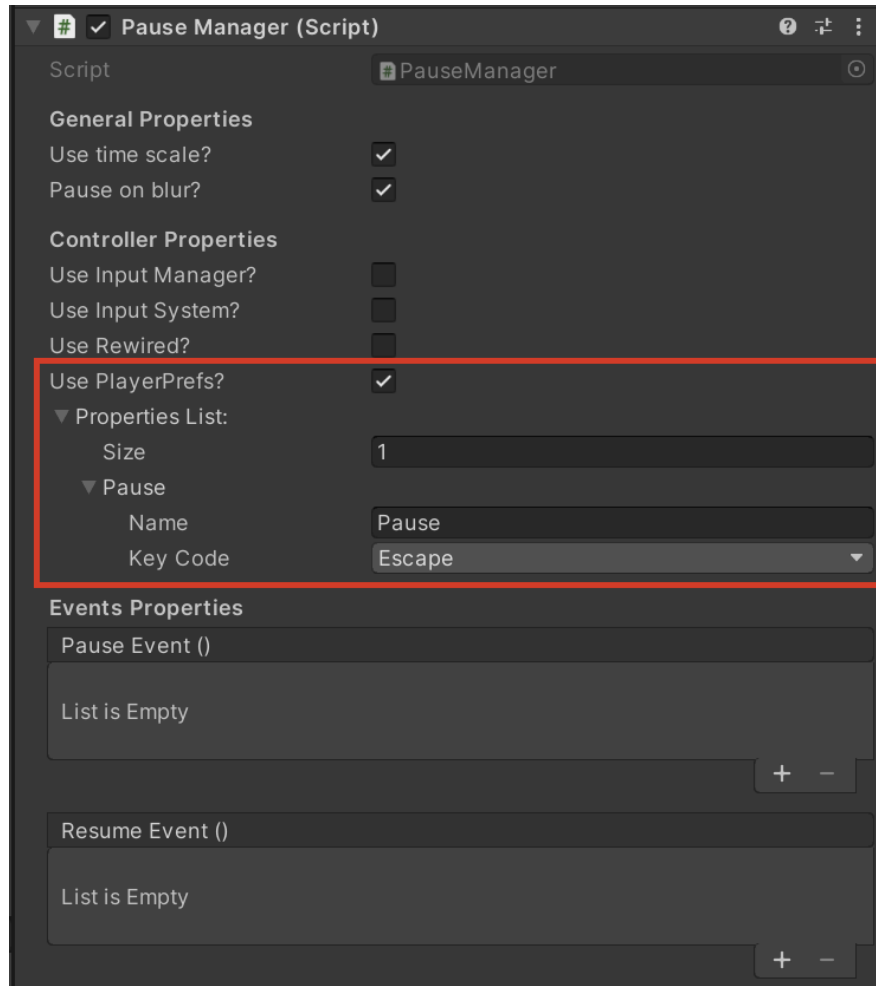
Resume Event ()

List is Empty

+ -

PlayerPrefs

Just in case you want to save/load the pause/resume **key** on **PlayerPrefs** and use it across all scenes, just check the **Use PlayerPrefs** property under **Controller Properties**.



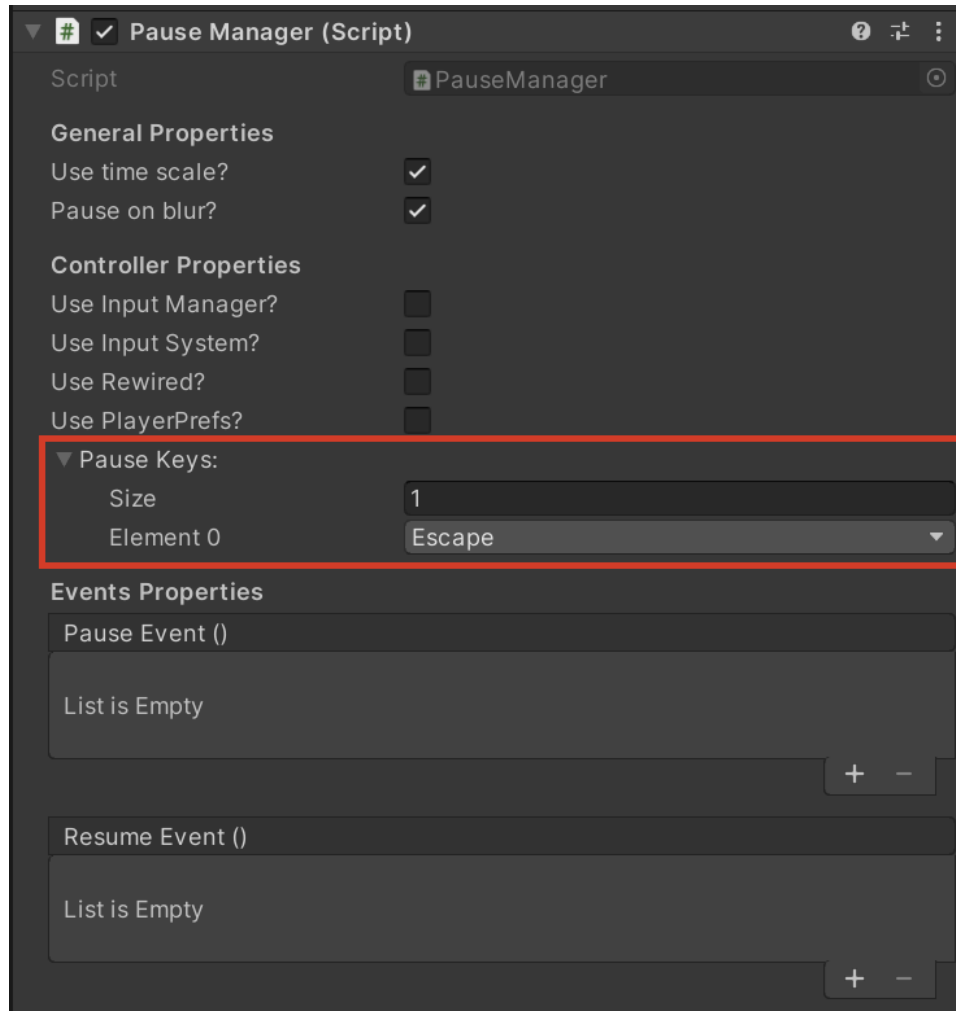
The property's name is used for recovering the **KeyCode** from **PlayerPrefs**.

You can define many properties for the list above, such as **Player 1 Pause**, **Player 2 Pause**, etc and the **KeyCode** associated with those properties are the **Escape** key.

By default, there's only one property on the list with the name **Pause** and the **Escape** key.

Pause Keys

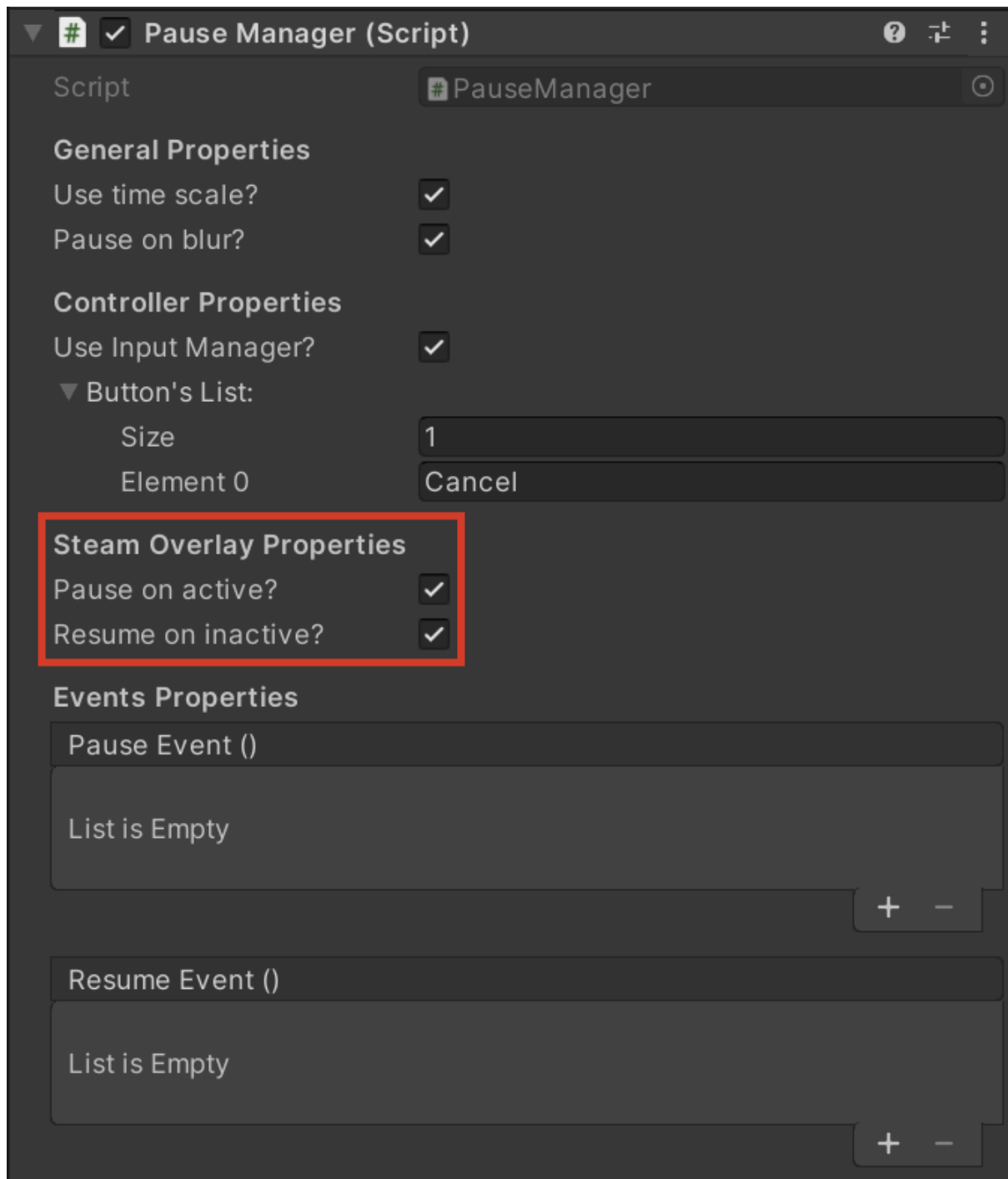
If none of the Controller Properties options suits you, you can use the **Pause Keys** list, which defines specific **KeyCode** to pause/resume.



By default, there's only one key, which is the **Escape** key.

Steam Overlay Properties

To publish a game on [Steam](#), it's mandatory that your game can pause automatically when the **Steam Overlay** window appears. To achieve this, simply install the [Steamworks.NET](#) package, which is free and integrated with **Pause Manager**, and the property to pause when **Steam Overlay** is active will appear.

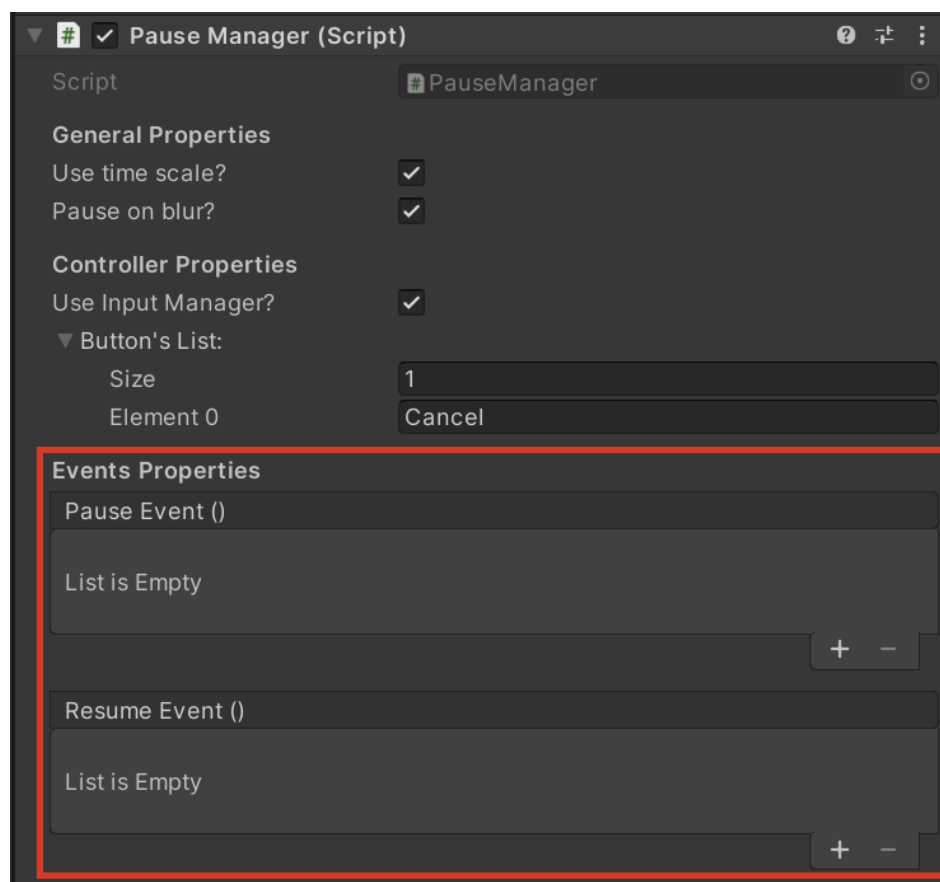


Whether your game should pause when the Steam **Overlay** window is active, just check **Pause on active?** field.

Whether your game should resume when the Steam **Overlay** window is inactive, just check **Pause on inactive?** field.

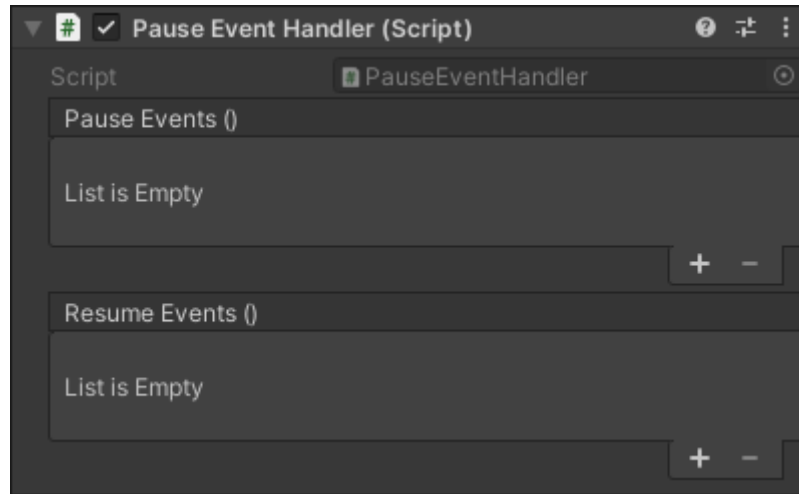
Events Properties

The events that are triggered when the game is paused or resumed. Usually used to show/hide a pause menu. Please check the **Example Pause Menu Scene** under **Assets/Gabriel Pereira/Pause Manager/Example** folder.



Pause Event Handler

When you decide **not** to use **Unity's time scale**, through **Use time scale** property under **General Properties**, there's a script called **Pause Event Handler** that can be attached to a **GameObject**. It has its pause/resume events for specific behaviors that can be defined when the game is paused/resumed (eg.: stop player's movement, stop projectile's movement, switch player to idle state, etc).



Examples

Check the **Example Scene** and **Example Pause Menu Scene** under **Assets/Gabriel Pereira/Pause Manager/Example** for more details.