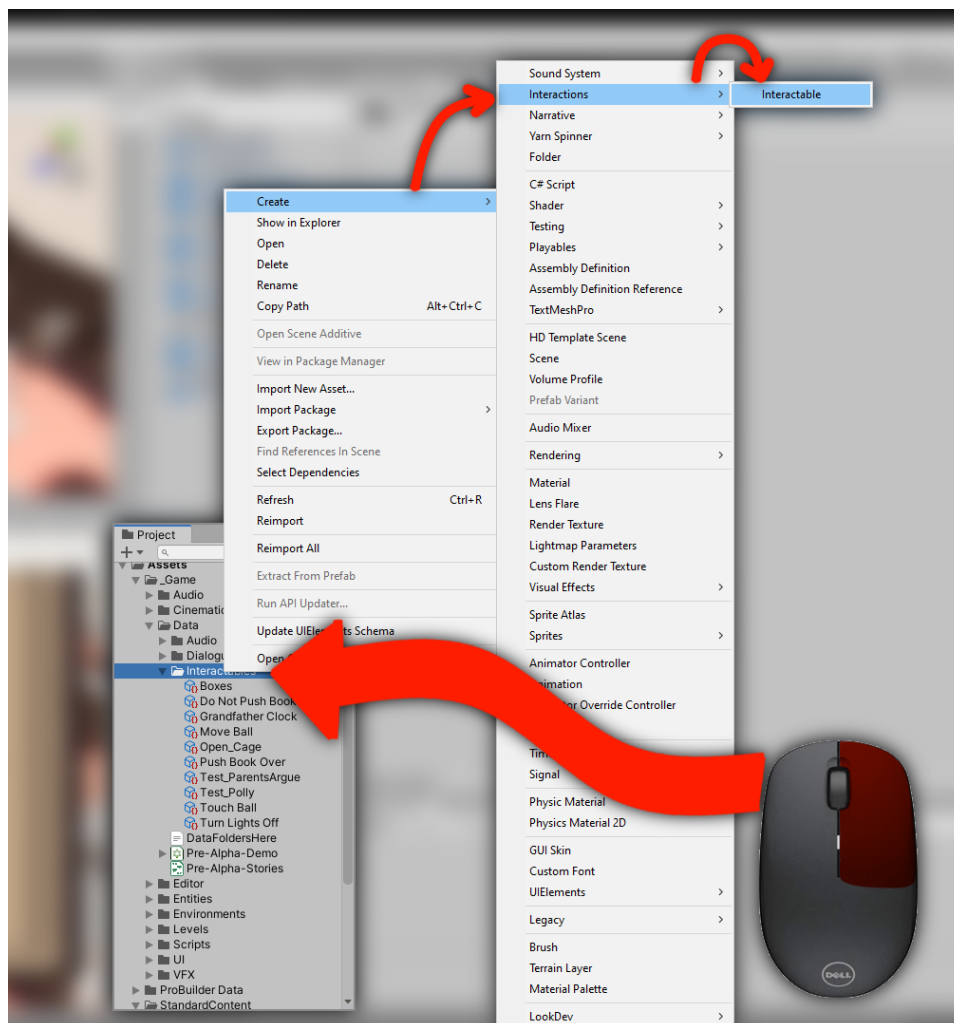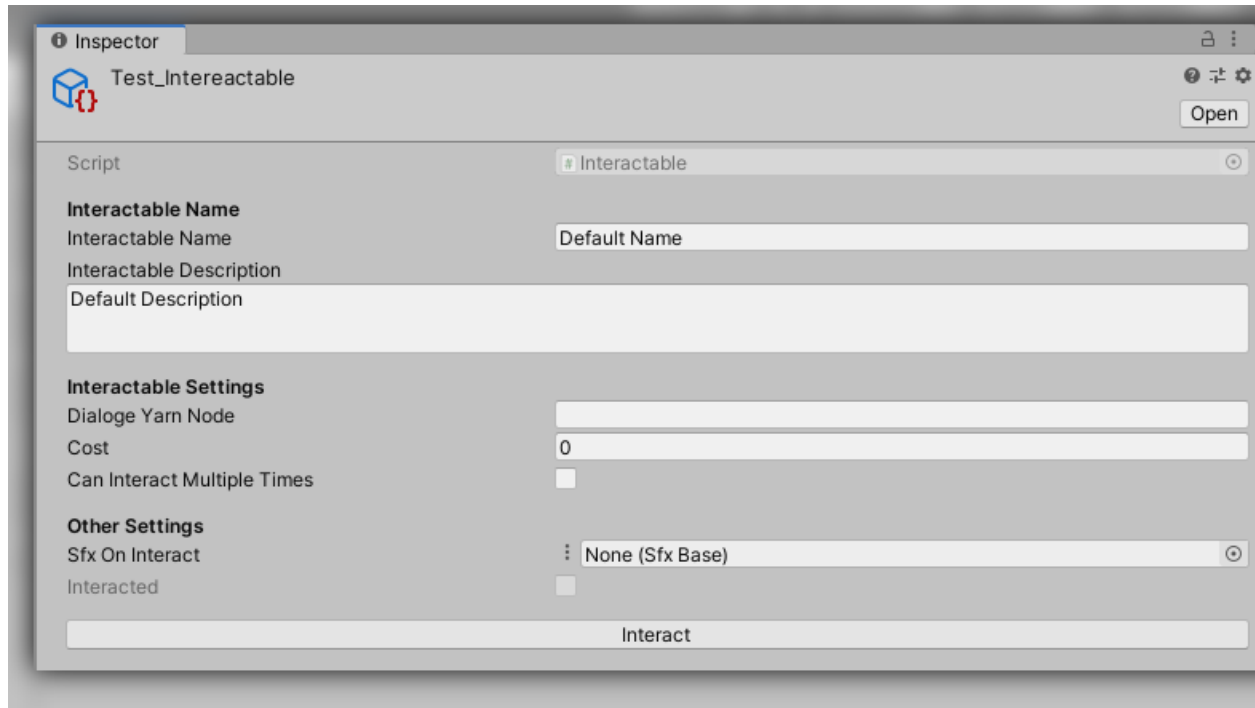# Interaction System Documentation

By: Sai Mangipudi

1.Creating an Interaction:

- Right click in the **interactables folder**
- Hover over **Create**
- Hover over **Interactions**
- And Click on **Interactable**

- **Interactable Name: IMPORTANT**
    - This is where you can set a custom name for the interaction.
    - This name does not affect the name of the actual object.
        - For example, I can type in "StankyBreath" into the **Interactable Name** slot, and the name of the Scriptable Object in the **_Assets** folder will still be **"Test_Interactable"**.
    - *RECOMMENDATION:* Keep some sort of system or naming convention for both the name of the Interaction in the _Assets folder and the name in the Interactable Slot. They can either be different or the same, but make them unique from other interactions.
    - *NOTE*: Please make the Interactable Name unique from other interactions! The Save and Load System depends on the name variable being different for each interaction!
- **Interactable Description:**
    - This is just a string variable that carries the description of the interaction. This variable will not affect any mechanic or any visual display. Instead this is just there to make it easier for Level Design or whoever makes interactions to keep track of what the interaction does.
- **Interactable Settings:**
   - For more information on Dialogue and Yarn Nodes, ask Nick (Sir

Hippolyte#0788 on discord).

- - **The Cost** is the number of Spirit points it would take to be able to run this interaction.
  - - **Can Interact Multiple Times:** Check this box if the interaction occurs multiple times in different seasons.
- - **Other Settings:**
  - - **SFX Base:** Feel free to attach a SFX Base file. Ask Brandon for more info (Brandon_#4378 on discord).
- - **Interact Button:** Press this to test the interaction after the Scriptable Object is hooked up to both the **Trigger Object** and the **Acting Object** (both these terms will be explained soon).

This is essentially how you would go about creating an interaction. Doing this process creates an interaction scriptable object. Bear in mind that you would need to do this process for each interaction that occurs. ***NOTE*: This process needs to be done for both the main interactions and any alternate interactions!**

2. Trigger Objects:

Trigger Objects are objects that are clicked on by the **Player** to cause an interaction event to occur. They have the **Story Interactable Component** on them. If a **Trigger Object** doesn't have the **Story Interactable Component**, feel free to add it.
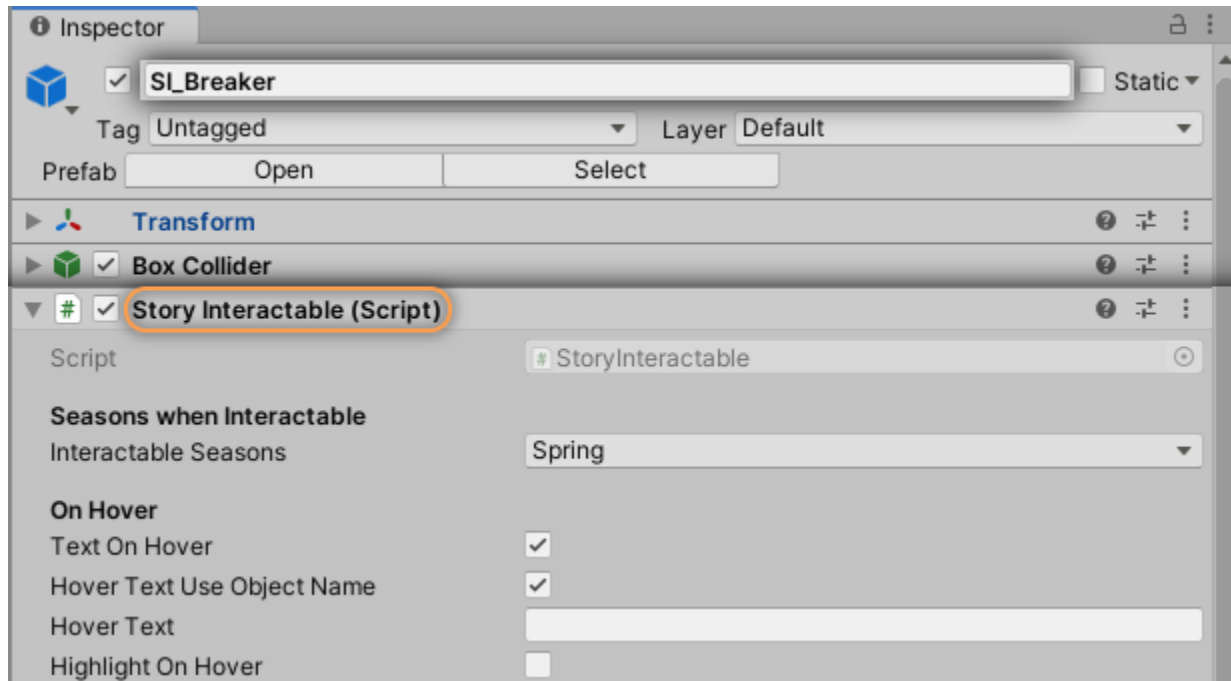
SI_Breaker is considered a **Trigger Object** because it will be clicked on by the Player and it has the Story Interactable Components.
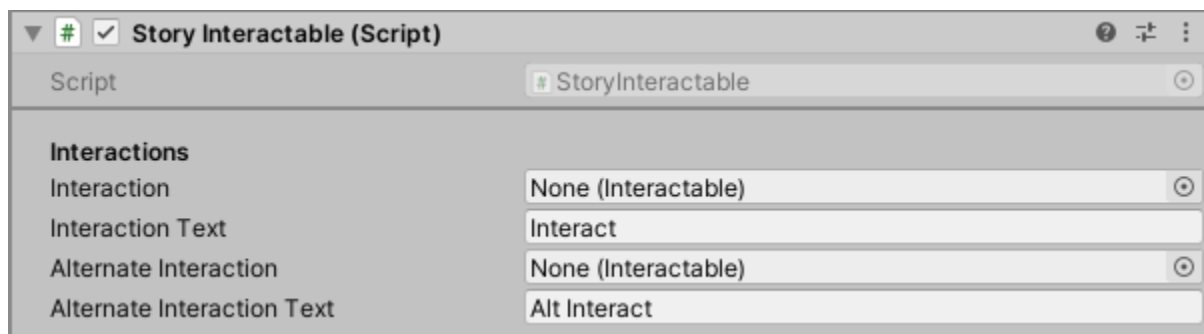
At the bottom of the **Story Interactable Component** is where you'd hook up the **Interactions** you've created in the previous step. If an Interaction Event only has one interaction, then leave the **Alternate Interaction** blank.

The **Interaction Text Fields** are where you can type the text that you'd like to appear

on the **Window** that pops up when the player clicks on the **Trigger Object**.
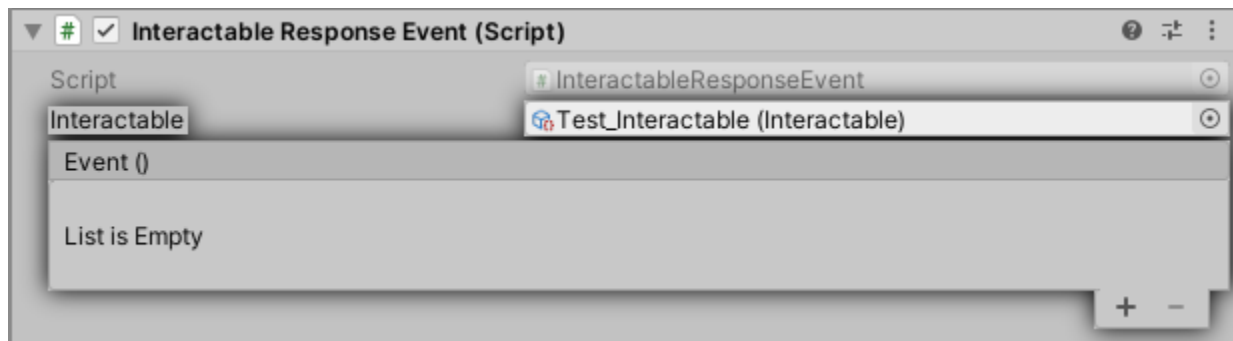
3. Acting Objects

Acting Objects are the objects that respond to the **Player's Click**. All **Acting Objects** will have the **Interactable Response Event Component** on them. If an **Acting Object** doesn't have the **Interactable Response Event Component,** feel free to add it there.
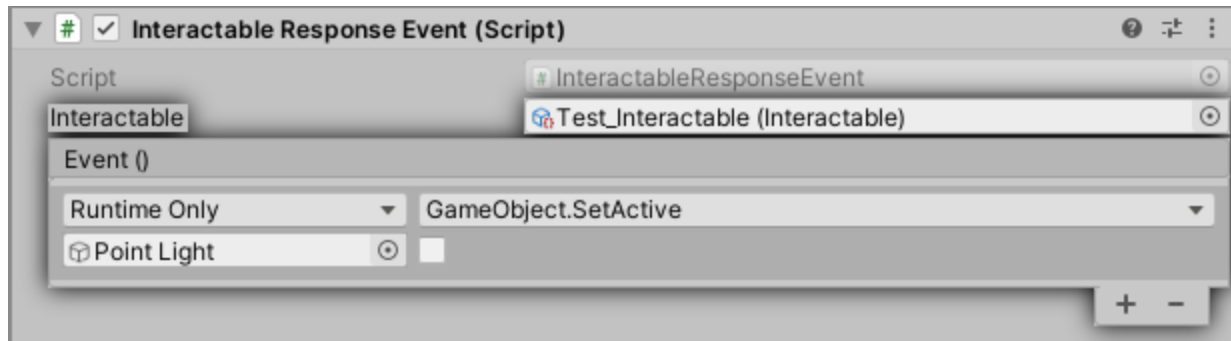

In this case, I have created a test **Point Light** that acts as an **Acting Object** for the **SI_Breaker** Object. I'm going to use this light as an example.
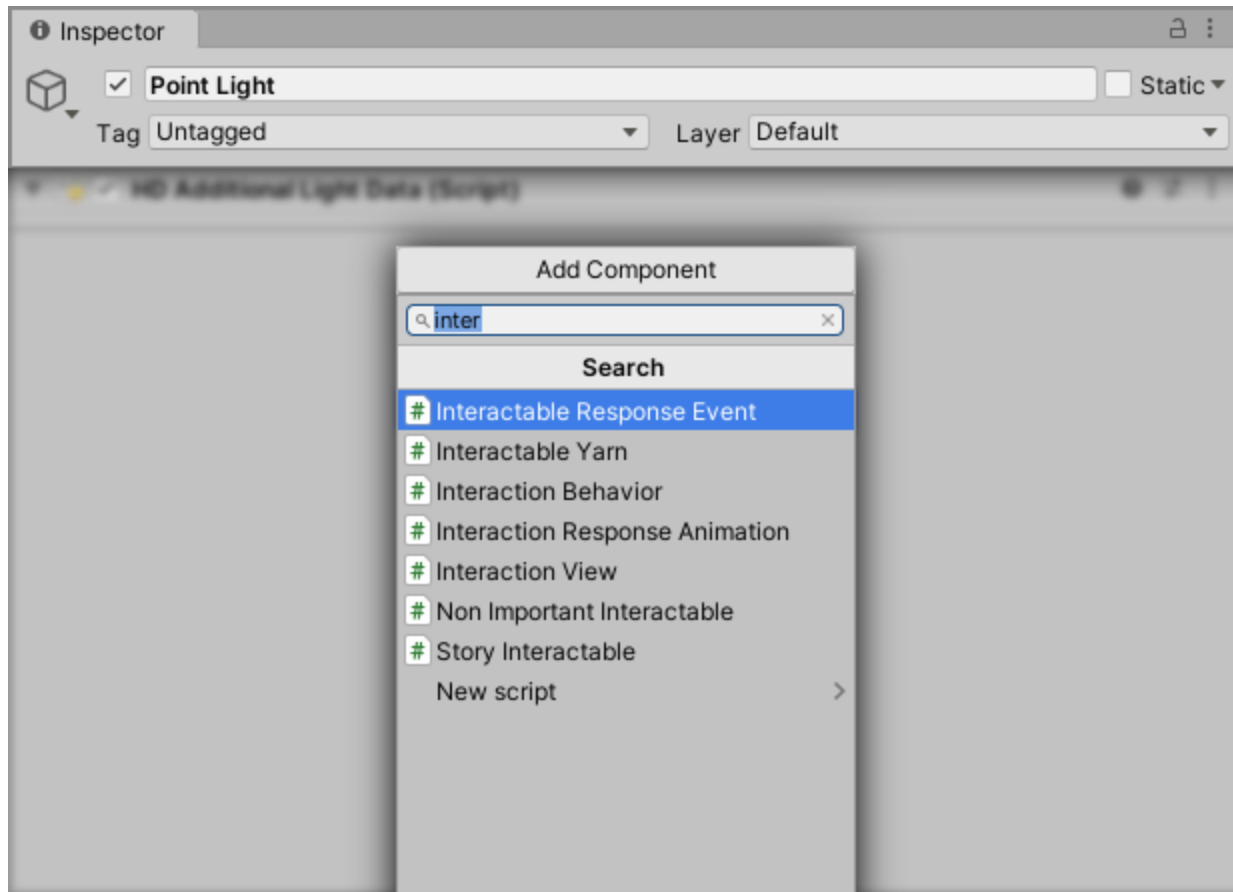
Since the Point Light doesn't have an **Interactable Response Event Component**, I'll go ahead and add one.

After that, you would go ahead and connect the Interaction you created in Step One into the **Interactable Response Component.**

Now, you would connect up whatever script controls the **Acting Object** into the **Event Box**, and set it up like you would set up a **Button** in Unity. The setup I have here, makes it so that the **Point Light** is disabled when the **Test_Interactable** is triggered. You can connect as many functions as you'd like. In fact, you can even connect separate objects. This event system is public so the **Interactable Response Event** could technically go on the **Trigger Object** and just trigger to other objects from there.

And that's it. However, there are quite a few connections going around. Please don't forget to hook up the **Interactable Scriptable Object** to both the **Acting Object** and **Trigger Object**.

NOTE: The Event Boxes do depend on Other Scripts being present to be able to connect. Or you can look through the already present object and the functions already on the objects. If you would like certain functionality on Trigger Objects, let me know (TheGeekSKM#8007 on discord). I will get the functionality ready asap.