

Challenge 1: Raising the Event

What is the event?

A UnityEvent is triggered when the player clicks the **ToggleButton**, **FlashButton**, or **LampSpawner** UI.

What's the logic for that?

- Use Button.onClick from Unity's UI system.
- Some listeners are added dynamically with AddListener() based on key press (KeyCode.S).
- Others are assigned in the Inspector.
- Lamp detects if it's broken (lampBreaker.broken) and changes visuals based on event.

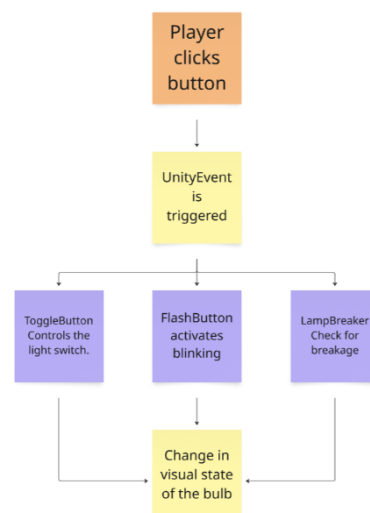
Pseudocode

```
if (Input.GetKeyDown(KeyCode.S))  
{  
    if (!listenerAdded)  
        button.onClick.AddListener(soundScript.Click);  
    else  
        button.onClick.RemoveListener(soundScript.Click);  
}
```

Who raises the event?

- Button click triggers event.
- LampBreaker and FlashButton both respond to it.
- LampFlasher uses coroutine when button event is raised.

Visual Flow



Challenge 2: Subscribing & Unsubscribing

Who subscribes to events?

- FlashButton script dynamically adds/removes listener to play sound.
- Manager-like logic subscribes LampFlasher to button click during “Flash Mode”.

Who unsubscribes?

- FlashButton.cs controls subscription with KeyCode.S toggle.

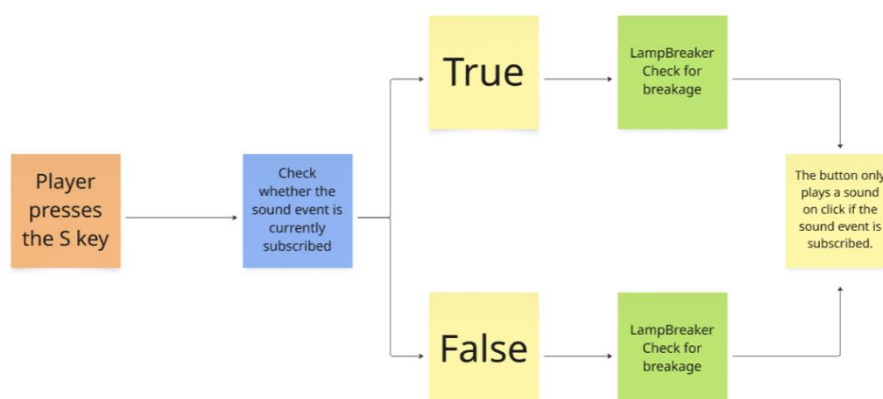
What's the logic to do this?

- Use AddListener() only if !listenerAdded
- Use RemoveListener() if listenerAdded
- Toggle between states to ensure event doesn't stack.

Pseudocode

```
if (Input.GetKeyDown(KeyCode.S))  
{  
    if (!listenerAdded)  
        button.onClick.AddListener(soundScript.Click);  
    else  
        button.onClick.RemoveListener(soundScript.Click);  
}
```

Visual Flow



Challenge 3: Coroutine Control

What is the coroutine?

- LampFlasher uses a coroutine FlashRoutine() to animate lamp scaling up/down in loop.
- Runs for 5.5 seconds.

When is it stopped?

If any of these conditions fail:

- toggleButton.isOn == false
- lampBreaker.broken == false
- Time runs out (finalTime >= 5.5f)

Pseudocode

```
while (Time < 5.5f && toggleButton.isOn && lampBreaker.broken)
{
    animate scale up/down
}
```

Key Concept

- Coroutine makes the animation smooth.
- Stops automatically when logic breaks or conditions change.

Visual Flow

