

# Red Point Notify System

The red point notify system is an observer design pattern. It's designed to be easy to use!

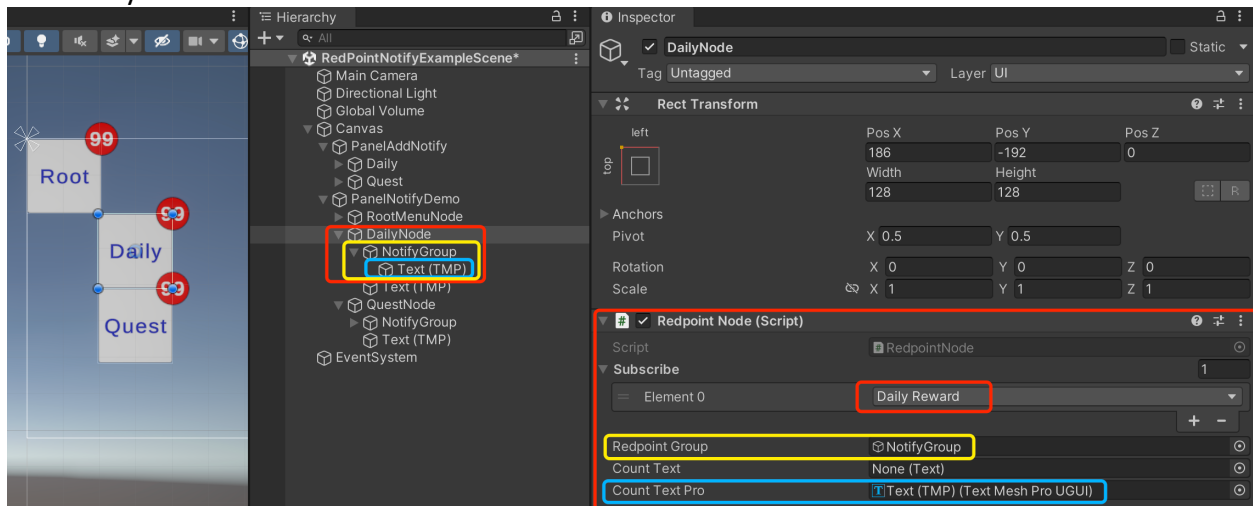
## 1. Define the item type

The red point item type is an Enum type, modify this enum to satisfy your purpose:

```
public enum ERedPointItem
{
    None,
    VersionUpdate,
    PlayerUpgrade,
    DailyReward,
    Quest,
    Achievement,
    Max
}
```

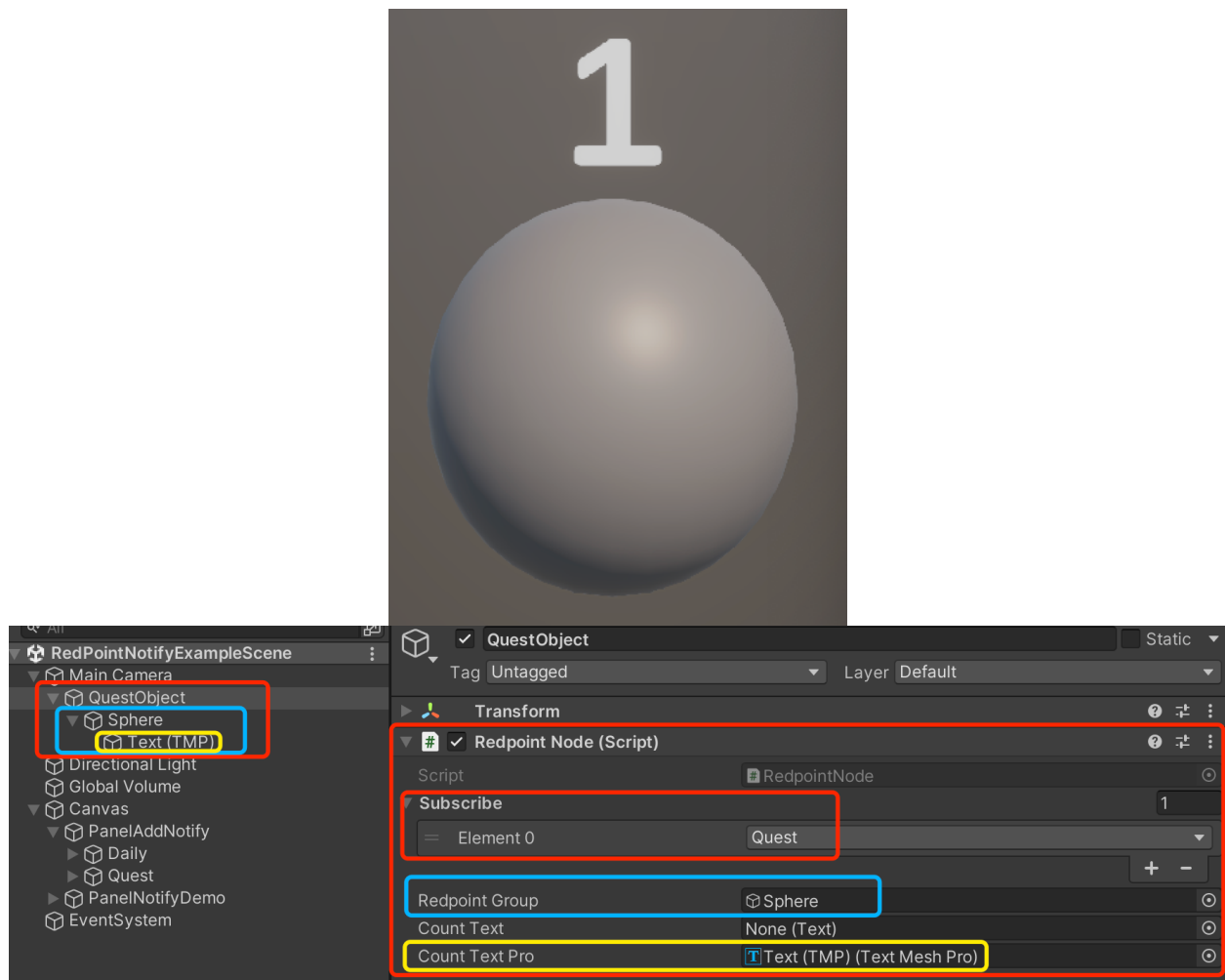
## 2. Redpoint Node

The notify subscribe node looks like below:



- **Subscribe List:** add any number of the red point item type to the list.
- **Redpoint Group:** the whole group will show or hide.
- **Count Text:** UnityEngine.UI.Text to show the total count.
- **CountTextPro:** TMPro.TMP\_Text to show the total count.

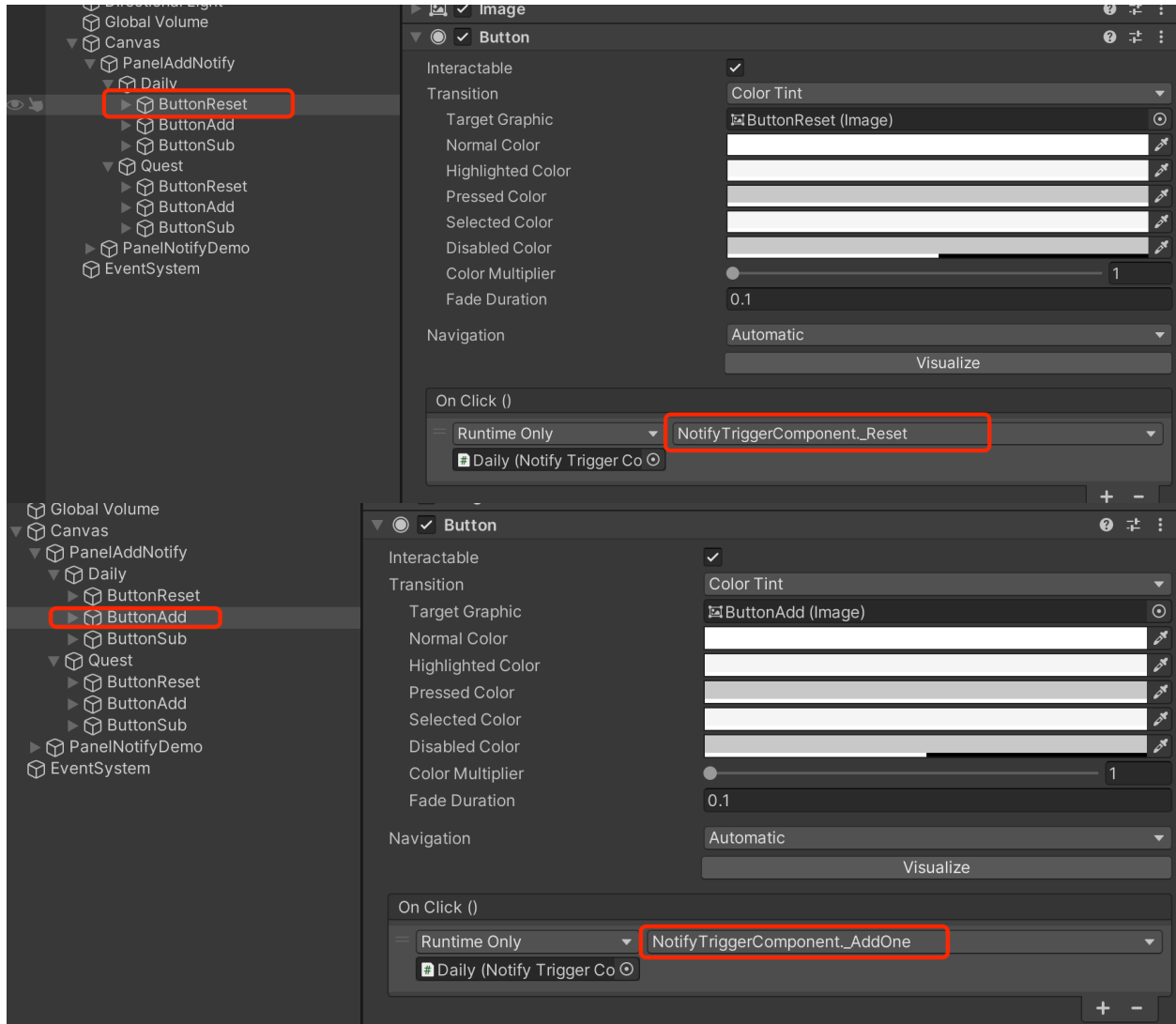
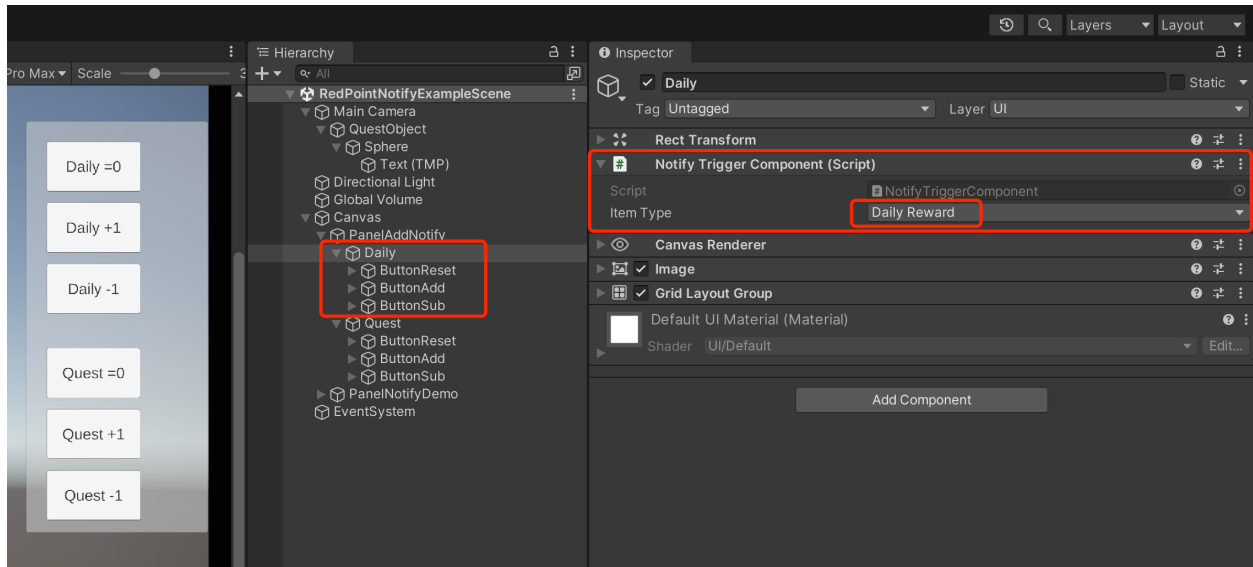
Another example in 3D objects, it's the same way to control the notify show or hide:



### 3. Notify Trigger Component

This is a helper component to access the RedPointyNotify api, such as add/sub/reset the count of a redpoint item type.

For example, the Daily buttons group will modify the DailyReward redpoint item count, each of the button bind to the helper component's function.



And the code implementation:

```
public class NotifyTriggerComponent : MonoBehaviour
{
    public ERedPointItem ItemType;

    public void _AddOne()
    {
        _Add(1);
    }

    public void _SubtractOne()
    {
        _Add(-1);
    }

    public void _Add(int count)
    {
        RedPointNotify.AddMark(ItemType, count);
    }

    public void _Reset()
    {
        RedPointNotify.ClearMark(ItemType);
    }
}
```