Extended Buttons

Unity doesn’t provide easy way to extend buttons behaviour. Especially for 3D objects. This plugin solve problems like this.

Features

- Button2D

  User can add listeners to 5 events
  - onEnter - invoke once when pointer enter button
  - onDown - invoke once when clicked down on button
  - onUp - invoke once when clicked up from button
  - onClick - invoke once when click down and up where invoked on the same button
  - onExit - invoke once when pointer exit button

  User can add listeners to events via inspector or via scripts like in UnityEngine.UI.Button onClick event

- Button3D

  User can add listeners to 5 events
  - onEnter - invoke once when pointer enter GameObject
  - onDown - invoke once when clicked down on GameObject
  - onUp - invoke once when clicked up from GameObject
  - onClick - invoke once when click down and up where invoked on the same GameObject
  - onExit - invoke once when pointer exit GameObject

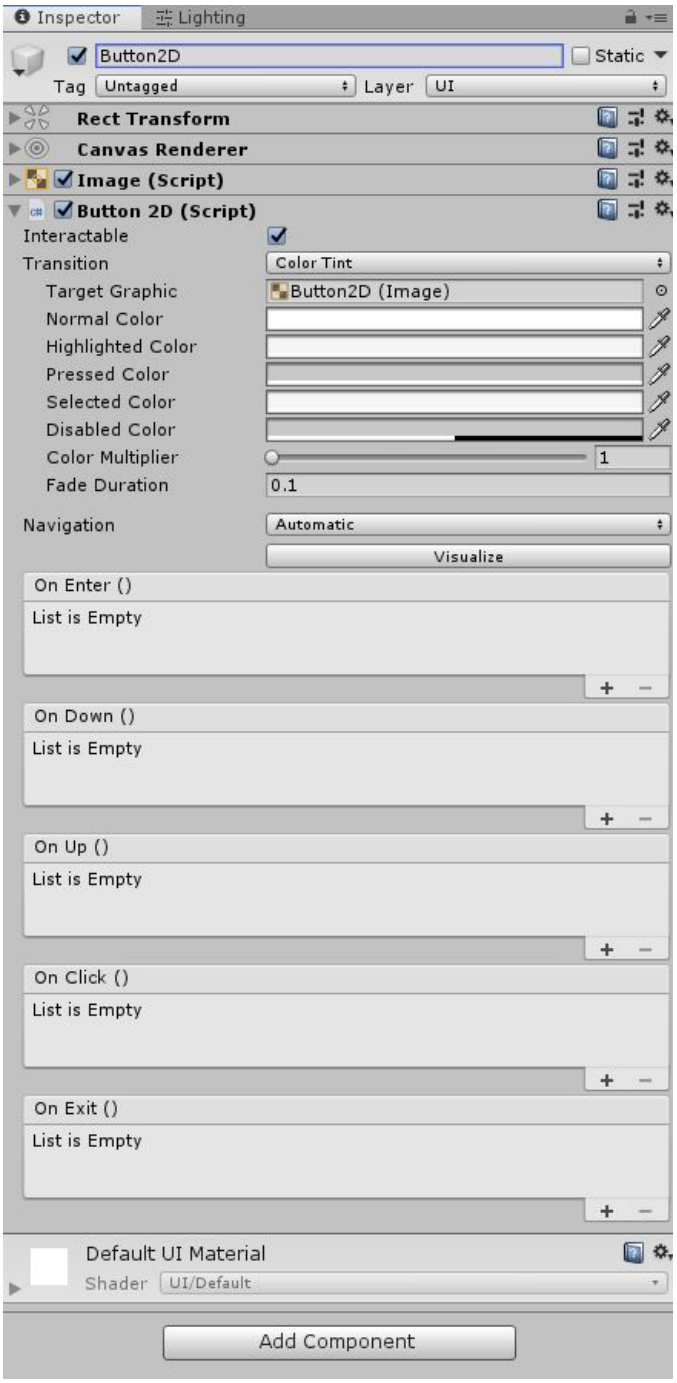
  User can add listeners to events via inspector or via scripts

Usage

Button2D

Add “Button2D” component to Canvas element then add listener / listeners to event / events you want. You can add listeners via inspector or script:

- Inspector
    - ***add*** - click plus “+” sign on event you want to add listener
    - ***remove*** - select listener and click minus “-” sign to remove selected listener



- Script
    - ***add*** - to add listener catch reference to Button2D then add listener to event you want:

```
button?.event.AddListener(YourMethod);
```

where:

- button - reference to Button2D
- event - one of 5 events (onEnter, onDown, onUp, onClick, onExit)
- YourMethod - method you want to add as listener

Example (when GameObject contains Button2D component):

```
var button = GetComponent<Button2D>();
button?.onEnter.AddListener(YourMethod);
```

```
[SerializeField] private Text debugText;

private void Start()
{
    var button = GetComponent<Button2D>();
    button?.onEnter.AddListener(() =>
    {
        Debug.Log("OnButton Enter");
        debugText.text = "OnButton Enter";
    });
    button?.onDown.AddListener(() =>
    {
        Debug.Log("OnButton Down");
        debugText.text = "OnButton Down";
    });
    button?.onUp.AddListener(() =>
    {
        Debug.Log("OnButton Up");
        debugText.text = "OnButton Up";
    });
    button?.onClick.AddListener(() =>
    {
        Debug.Log("OnButton Click");
        debugText.text = "OnButton Click";
    });
    button?.onExit.AddListener(() =>
    {
        Debug.Log("OnButton Exit");
        debugText.text = "OnButton Exit";
    });
}
```

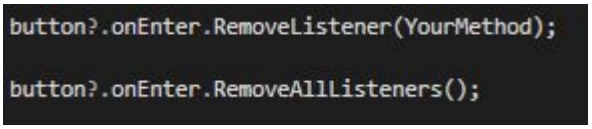
- ***remove*** - to remove listener use method from UnityEvent class: RemoveListener(UnityAction call) or RemoveAllListeners()

  to remove one listener

```
button?.onEnter.RemoveListener(YourMethod);
```

  or to remove all listeners

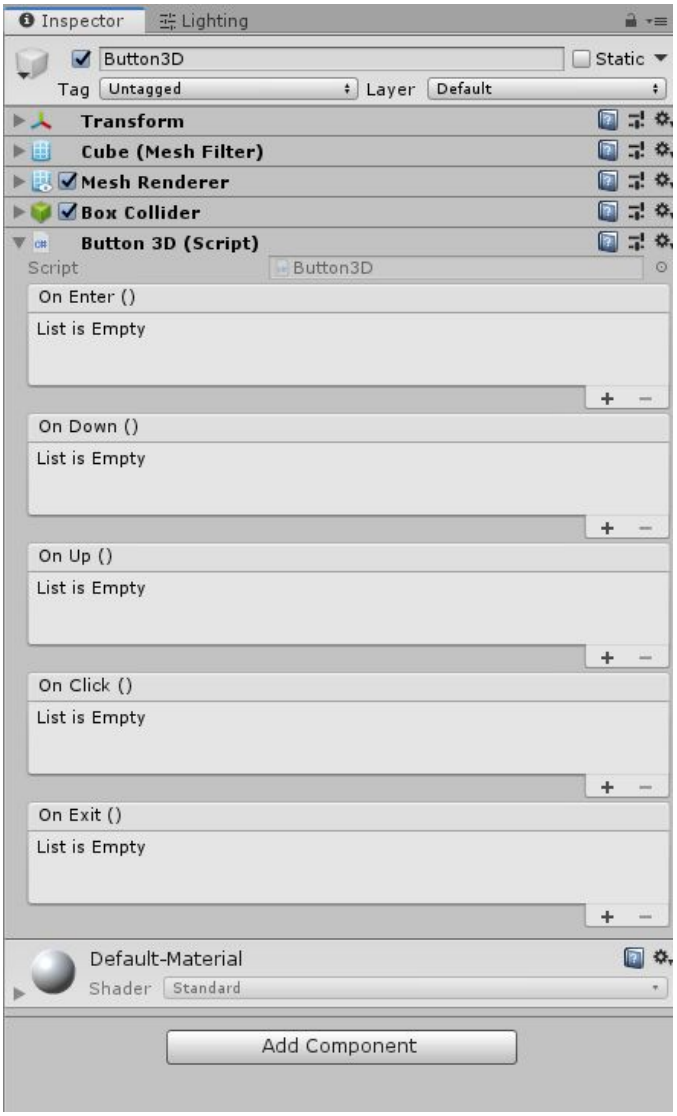
```
button?.onEnter.RemoveAllListeners();
```



Button3D

Add “Button3D” component to GameObject on scene then add listener / listeners to event / events you want. Remember that GameObject require Collider to detect interaction. You can add listeners via inspector or script:

- Inspector
    - ***add*** - click plus “+” sign on event you want to add listener
    - ***remove*** - select listener and click minus “-” sign to remove selected listener



- Script
  - ***add*** - to add listener catch reference to Button3D then add listener to event you want:

```
button?.event.AddListener(YourMethod);
```

where:
- button - reference to Button3D
- event - one of 5 events (onEnter, onDown, onUp, onClick, onExit)
- YourMethod - method you want to add as listener

Example (when GameObject contains Button3D component):

```
var button = GetComponent<Button3D>();
button?.onEnter.AddListener(YourMethod);
```

```
[SerializeField] private Material red;
[SerializeField] private Material white;
[SerializeField] private GameObject child;

private void Start()
{
    var meshRenderer = GetComponent<MeshRenderer>();
    var button = GetComponent<Button3D>();
    button?.onEnter.AddListener(() =>
    {
        Debug.Log("OnButton Enter");
        meshRenderer.material = red;
    });
    button?.onDown.AddListener(() =>
    {
        Debug.Log("OnButton Down");
        child.SetActive(true);
    });
    button?.onClick.AddListener(() =>
    {
        Debug.Log("OnButton Click");
    });
    button?.onUp.AddListener(() =>
    {
        Debug.Log("OnButton Up");
        child.SetActive(false);
    });
    button?.onExit.AddListener(() =>
    {
        Debug.Log("OnButton Exit");
        meshRenderer.material = white;
    });
}
```

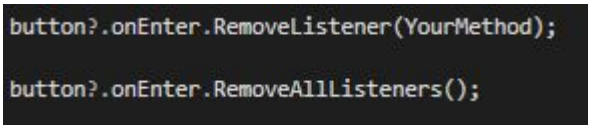
- ***remove*** - to remove listener use method from UnityEvent class: RemoveListener(UnityAction call) or RemoveAllListeners()

  to remove one listener

```
button?.onEnter.RemoveListener(YourMethod);
```

  or to remove all listeners

```
button?.onEnter.RemoveAllListeners();
```



To detect interaction with Button3Ds you must place on scene prefab with class derived from IButtonsListener. Package contains ready to use prefab (“ButtonsListenerBasic”) in folder “Prefabs”.

You can create your own ButtonsListener, IButtonsListener provides Listener() method where you can write your own logic how to process interaction between input and Button3D.

Summary

Package contains logic and example scripts, sample scenes.

To manage 3D buttons place prefab with listener on scene ("ButtonsListenerBasic" in folder “Prefabs”) then add to GameObject with Collider component "Button3D". ButtonsListenerBasic is only for 3D Buttons. Button2Ds don’t need that.

To extend UI buttons replace Button component with "Button2D".

GitHub: https://github.com/ostrzolekpawel/UnityExtendedButtons