Adding Basic Button Functionality in Unity 6

Overview

Buttons are essential UI elements that allow users to interact with applications. In this tutorial, you will implement basic button functionality using Unity's Event System. By the end, you will be able to:

- Understand the role of the EventSystem GameObject in interactive UIs.
- Identify use cases for buttons in various UIs.
- Implement simple button interactions using the Event System.

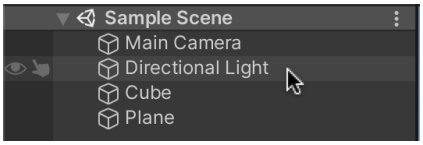
---

1. Understanding UI Feedback Mechanisms

A well-designed UI should clearly communicate with users through visual feedback. Common feedback cues include:

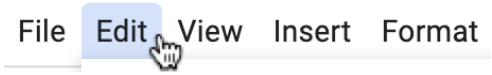
- **Hovering:** Buttons change color slightly when hovered over.

Hovering over the Edit button in Google Docs

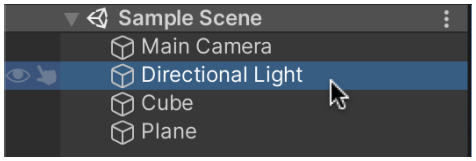


Hovering over a GameObject in the Unity Hierarchy.

- **Clicking:** Buttons change color noticeably when pressed.
- **Selection:** The selected button remains highlighted.



Selecting the Edit button in Google Docs



Selecting a GameObject in the Unity Hierarchy.

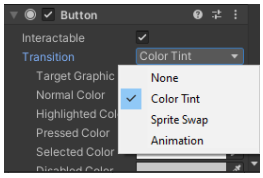
Try this: Hover over buttons in Unity or other applications and observe how they change.

---

2. Editing Button Transition Colors

Unity allows you to modify button color transitions to enhance user feedback.

1. **Select the Button:** In the Hierarchy, choose the **Settings Button** GameObject.



2. **Locate the Button Component:** In the Inspector, find the **Transition** property.
3. **Set Transition Type:** Ensure **Color Tint** is selected.
4. **Modify Colors:** Adjust the following color properties:
    - **Normal Color:** Default white (no tint).
    - **Highlighted Color:** Slightly darker for hover effect.
    - **Pressed Color:** Noticeably darker.
    - **Selected Color:** Not relevant in this case.
    - **Disabled Color:** Not needed as the button is always enabled.

5. **Adjust Color Multiplier:** Increase if your button is dark or semi-transparent.
6. **Test in Play Mode:** Observe how the button responds to interactions.

---

3. Adding an Action to the On Click Event

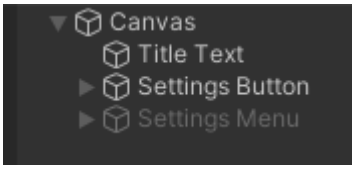
Buttons perform actions when clicked using UnityEvents.

1. **Select the Button:** Choose the **Settings Button** in the Hierarchy.
2. **Locate On Click () in the Inspector:** This section is empty by default.
3. **Add an Action:** Click the **+** button to add a new event.

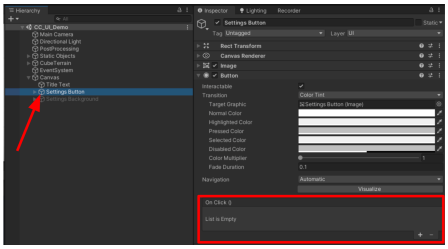
---

4. Making the Settings Menu Appear

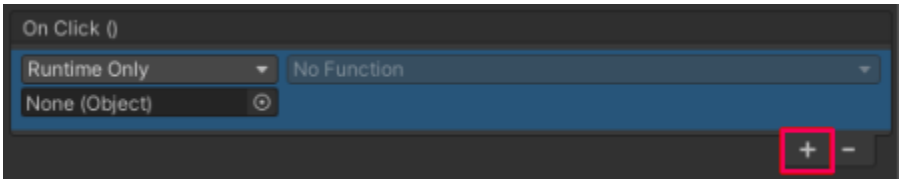
1. **Ensure Proper Object States:**
    - Activate **Title Text** and **Settings Button** GameObjects.
    - Deactivate **Settings Menu** GameObject.



2. **Assign the Target Object:**
    - Drag **Settings Menu** into the On Click () event.



3. **Select a Function:**
    - From the dropdown, choose **GameObject > SetActive (bool)**.

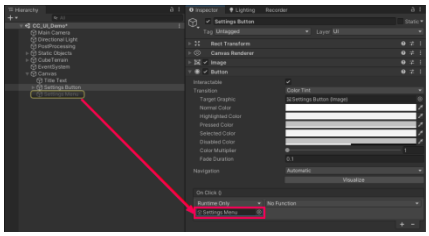


4. **Enable the Checkbox:** This calls **SetActive(true)**, making the settings menu appear.
5. **Test the Interaction:** Clicking the settings button should now display the menu.

---

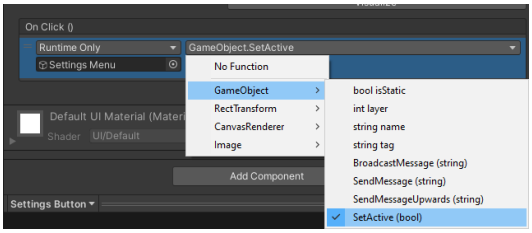
5. Hiding the Title Screen Elements

1. **Select the Settings Button.**

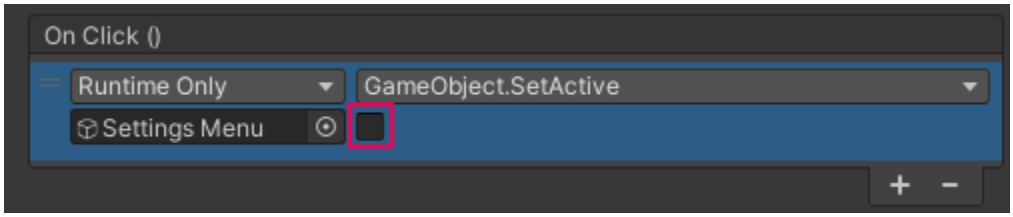


2. **Add Two More Actions:** Click **+** twice in the On Click () section.
3. **Assign Objects:** Drag **Title Text** and **Settings Button** into the new actions.

4. **Disable Visibility:**
    - Set the function to **GameObject > SetActive (bool)**.
    - Uncheck the checkbox (calls **SetActive(false)**).



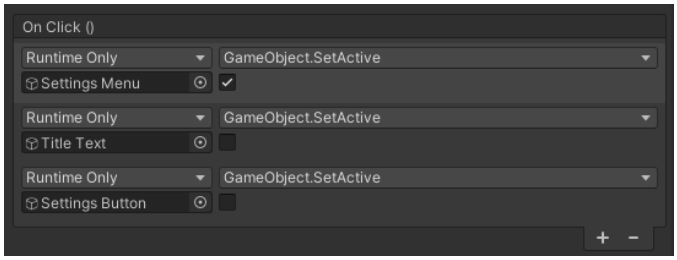
5. **Test in Play Mode:** The settings menu should now replace the title screen elements.



---

6. Navigating Back to the Title Screen

1. **Select the Exit Button in the Settings Menu.**
2. **Add Three Actions:** Click **+** three times in On Click ().
3. **Assign Objects:** Drag **Settings Menu, Title Text,** and **Settings Button** into the actions.
4. **Set Visibility Functions:**
    - **Settings Menu: SetActive(false)** (disappears).
    - **Title Text & Settings Button: SetActive(true)** (reappear).



5. **Test the Interaction:** The exit button should now restore the title screen.