

UIController Reference

Links

- [Asset Store](#)
- [GitHub](#)
- [Forum](#)
- [Support Email](#)

Videos

- [Examples](#)
- [Tutorial](#)

Tutorial

1. Add UIController component to GameObject, Animator will be create.
2. Create Override Controller. ("Project" window > Create > Animator Override Controller)
3. Click "Quick Setup/Show_Hide" on Inspector window.
4. Drag controller to UIController Animator "Controller" Variable.
5. Edit the UI animation on Animation window.
6. Click "Show / Hide" On UIController component with Playing Mode to test animations.

Variables

- showOnAwake
 - If enabled, show animation will start playing when GameObject enable.
- onHideAction
 - A enum value of action to on hide
 - None: No action.
 - Disable: Disable GameObject.
 - Destory: Destory GameObject.
- isShow
 - A bool value of state.
 - true: "Show" animation playing or end.
 - false: "Hide" animation playing or end.
- isPlaying
 - A bool value of animation state.
 - true: "Show" or "Hide" animation is playing.
 - false: "Show" or "Hide" animation is end.
- animator
 - The animator the UI controller is attached to.

Events

- onShow
 - A UnityEvent that is invoked when "Show" animation end.
- onHide
 - A UnityEvent that is invoked when "Hide" animation end.

Public Functions

- Show
 - Play "Show" animation.
- Hide
 - Play "Hide" animation.

Code Examples

- Play "Show" animation

```
[RequireComponent(typeof(UIController))]  
public class MyPanel : MonoBehaviour {  
    void Start() {  
        this.GetComponent<UIController>().Show();  
    }  
}
```

```
public class MyPanel : UIController {  
    void Start() {  
        this.Show();  
    }  
}
```

- Add "Show" animation end event

```
public class MyPanel : UIController {  
    void Start() {  
        this.onShow.AddListener(() => {  
            print("Show animation end.");  
        });  
    }  
}
```