

BUI A box based basic UI system

By LukeWaffel

Instruction manual

Index

Just so you won't get lost in this "massive" document.

| | |
|--------------------------|----|
| Index | 2 |
| Basic usage instructions | 3 |
| Video tutorials | 8 |
| Documentation | 9 |
| Contact | 12 |

Basic usage instructions

Can it get any easier than this?

Using BUI is super easy! Easier than anything you've ever seen before!

Start off by adding the following line of code to the script you'll be using BUI in.

```
using LukeWaffel.BUI;
```

I will now walk you through creating your first UI box.

We'll start off by creating a new `UIBox` object. This will require two parameters. The first one being the unique ID for this box and the second one being one of the four default types (`Message`, `Warning`, `Success` or `Error`) or the name of a custom box that you have created (Note that this custom box **MUST** be located in `BUI/Resources/CustomBoxes`).

```
UIBox myFirstBox = new UIBox("firstBoxID", BUI.UIType.Message);
```

We'll now change the title and message of the box using

```
myFirstBox.header = "My first title!";
```

and

```
myFirstBox.body = "This is my first ever message written on my first ever box underneath my first ever title!";
```

By **LukeWaffel**

Now that that's done we can add some custom buttons if we like. A button requires a text to display on it and a function to call when it is pressed. This function needs to have two parameters, `UIBox boxInfo` and `UIButton buttonInfo`. The second variable contains information about the button that was pressed and the first variable contains information about what box this button was on.

We'll first create a new function which will Log some text to the console. Then we'll add a button to the box. (This function should be created outside the function you're in right now.)

```
void MyButtonFunction(UIBox boxInfo, UIButton buttonInfo){  
  
    Debug.Log("Button: " + buttonInfo.buttonText + " was pressed on box: "  
+ boxInfo.id);  
  
}
```

Now let's add a button to the box

```
myFirstBox.buttons.Add(new UIButton("First Button", MyButtonFunction);
```

If you want, you can add box-specific callback functions to your UI boxes. These functions will get called when the box you set them for is opened or closed, depending on the callback. You can do this using the following code.

First we need to create some functions again. One for closing, and one for opening. (This should be done outside of the function you're in right now. Somewhere near the button function)

```
void WhenMyBoxOpens(BoxDataCombo dataCombo){  
  
    Debug.Log("Box: " + dataCombo.boxData.id+ " was openend");  
  
}
```

By **LukeWaffel**

```
void WhenMyBoxCloses(BoxDataCombo dataCombo){  
  
    Debug.Log("Box: " + dataCombo.boxData.id+ " was closed");  
  
}
```

Now we need to assign these functions as the callbacks for our boxes. Luckily, this is super easy!

```
myFirstBox.onOpenedCallback = WhenMyBoxOpens;  
myFirstBox.onClosedCallback = WhenMyBoxCloses;
```

The first line will set the callback for when the box is opened, and the second line will set the callback for when the box is closed.

The only thing left now is to show the box right? Let's add it to the queue system. BUI will then handle the rest for you. Since BUI is a **singleton** you can easily access it using `BUI.Instance`.

```
BUI.Instance.AddToQueue(myFirstBox);
```

By **LukeWaffel**

That's it! You're done! You've made your first UI box. The final code should look something like this. Note that you can obviously use BUI in any function you want and not just start.

```
using LukeWaffel.BUI;
```

```
Void Start(){
```

```
    UIBox myFirstBox = new UIBox("firstBoxID", BUI.UIType.Message);
```

```
    myFirstBox.header = "My first title!";
```

```
    myFirstBox.body = "This is my first ever message written on my first ever box  
underneath my first ever title!";
```

```
    myFirstBox.buttons.Add(new UIButton("First Button", MyButtonFunction));
```

```
    myFirstBox.onOpenedCallback = WhenMyBoxOpens;
```

```
    myFirstBox.onClosedCallback = WhenMyBoxCloses;
```

```
    BUI.Instance.AddToQueue(myFirstBox);
```

```
}
```

```
void MyButtonFunction(UIBox boxInfo, UIButton buttonInfo){
```

```
    Debug.Log("Button: " + buttonInfo.buttonText + " was pressed on box: " +  
boxInfo.id);
```

```
}
```

```
void WhenMyBoxOpens(BoxDataCombo dataCombo){
```

```
    Debug.Log("Box: " + dataCombo.boxData.id+ " was openend");
```

```
}
```

By **LukeWaffel**

```
void WhenMyBoxCloses(BoxDataCombo dataCombo){
```

```
    Debug.Log("Box: " + dataCombo.boxData.id+ " was closed");
```

```
}
```

By **LukeWaffel**

Video tutorials

Because reading is boooooooooorriinnnggg

If you'd like to know how to create custom boxes and buttons check out the YouTube video's linked below.

[Basic box creation](#)

[Creating custom boxes and buttons](#)

Documentation

For the real nerds

(OBSOLETE) `public void SetOpenedCallback(OnBoxStateChangedEvent callback)`

This function sets the box opened callback

(OBSOLETE) `public void ResetOpenedCallback()`

This function resets the box open callback

(OBSOLETE) `public void SetClosedCallback(OnBoxStateChangedEvent callback)`

This function sets the boxed closed callback

(OBSOLETE) `public void ResetClosedCallback()`

This function resets the box closed callback

`BUI.boxOpenedCallack += MyCallbackFunction`

Subscribe to the boxOpenedCallback

`BUI.boxOpenedCallack -= MyCallbackFunction`

Unsubscribe from the boxOpenedCallback

`BUI.boxClosedCallack += MyCallbackFunction`

Subscribe to the boxClosedCallback

`BUI.boxClosedCallack -= MyCallbackFunction`

By **LukeWaffel**

Unsubscribe from the boxClosedCallback

```
public void AddToQueue(UIBox box)
```

This function is used to add a box to the queue

```
public void RemoveFromQueue(UIBox box)
```

This function is used to remove a box from the queue by a UIBox variable

```
public void RemoveFromQueue(string id)
```

This function is used to remove boxes from the queue using their ID

```
public void CloseBox(UIBox box)
```

This function closes a UIBox using a UIBox variable

```
public void CloseBoxes(UIBox[] boxes)
```

This function closes multiple UI boxes

```
public void CloseBoxes(string[] ids)
```

This function closes multiple UI boxes

```
public void CloseBox(string boxID)
```

This function closes a UIBox using the ID of the box

```
public UIBox[] GetActiveBoxData()
```

This function returns the data of the active boxes

```
public BoxDataCombo[] GetActiveBoxes()
```

This function returns all the ActiveBox objects

Contact

Because spamming developers is fun!

| | |
|------------------------------|------------------------|
| Website | lukewaffel.weebly..com |
| Email | lukewaffel@gmail.com |
| Twitter | @lukewaffel |
| Unity Forums Username | LukeWaffel |