



DAILY (TIME BASED) REWARDS



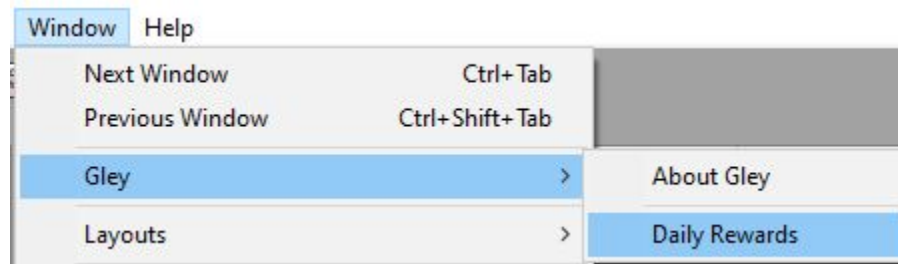
1. WHY DO YOU NEED TO USE THIS PLUGIN

- Increase the retention of your game by using Daily Rewards and Time Based rewards.
- Contains Timer buttons and Calendar popup rewards.
- Easy to use and display with a single line of code.
- Automatically adjusts resolution for both Landscape and Portrait mode.
- All setup is done in a Settings Window.
- Supports any number of days.
- Customizable time between consecutive rewards.
- First day can be open or not by default, based on settings.
- Full commented code, popups and demo scenes included.
- Works with Unity 2017.1 and above with Free, Plus or Pro license.
- Works on all supported Unity platforms.



2. SETUP GUIDE

- Import **Gley Daily Rewards Plugin** into Unity.
- Go to **Window->Gley->Daily Rewards** to open the Settings Window.



- Settings Window will open

Daily Rewards Settings Window - v.1.0.0

TIMER BUTTON SETUP:

Button ID (Unique): RewardButton

Time to pass: h: 0 m: 1 s: 0

Unlocked at start: ☒

Text after complete: Open

Interactable when unavailable: ☐

Remove Timer Button

Add Timer Button

CALENDAR SETUP:

Calendar Prefab: CalendarPopup

Calendar Canvas: CalendarPopupCanvas

First day unlocked: ☐

Restart at end of days: ☒

Time to pass: h: 0 m: 0 s: 15

Day 1

Texture: DR_CoinPlate

Value: 10

Remove Day

Day 2

Texture: DR_CoinPlate

Value: 20

Remove Day

Daily Rewards Settings Window - v.1.0.0

Remove Day

Day 8

Texture: DR_CoinPlate

Value: 1500

Remove Day

Day 9

Texture: DR_CoinPlate

Value: 2500

Remove Day

Day 10

Texture: DR_CoinPlate

Value: 5000

Remove Day

Day 11

Texture: Logo

Value: 10000

Remove Day

Add Calendar Day

Save

Open Timer Button Example

Open Calendar Example



Setup Timer Button

TIMER BUTTON SETUP:

Button ID (Unique)	RewardButton3		
Time to pass:	h: 4	m: 0	s: 30
Unlocked at start	<input type="checkbox"/>		
Text after complete	Open		
Interactable when unavailable	<input type="checkbox"/>		
<button>Remove Timer Button</button>			
<button>Add Timer Button</button>			

- **Button ID** -> a unique id for each button, it needs to be assigned on timer button script
- **Time to pass** -> time to pass for the reward to be available
- **Unlocked at start** -> if clicked the button starts as available and only after click the timer starts
- **Text after complete** -> text for the button when timer is 00:00:00
- **Interactable when unavailable** -> button can be clicked even when timer is running.
- **Remove Timer Button** -> deletes the current timer button
- **Add Timer Button** -> creates a new timer button



Setup Calendar Popup

CALENDAR SETUP:

Calendar Prefab ○

Calendar Canvas ○

First day unlocked ☐

Restart at end of days ☒

Time to pass: h: m: s:

Day 1

Texture ○

Value

Day 2

Texture ○

Value



Setup Calendar Popup

- **Calendar Prefab** -> the prefab that contains the calendar popup. Is located inside:
GleyPlugins/DailyRewards/Prefabs/CalendarPopup
- **Calendar Canvas** -> the prefab that contains the popup canvas. Is located inside:
GleyPlugins/DailyRewards/Prefabs/CalendarPopupCanvas
- **First day unlocked** -> if checked the first day is available from start, without the required time to pass
- **Reset at end of days** -> if checked after the last day is unlocked it will start from the beginning
- **Time to pass** -> time for unlocking the next day

Day setup:

- **Texture** -> the texture from the day prefab - should be representative for your rewards
- **Value** -> the reward amount



3. USER GUIDE - TIMER BUTTON

Subscribe to click listener from any script so you can grant and store the reward

- **GleyDailyRewards.TimerButton.AddClickListener(RewardButtonClicked);**

// **buttonID** -> the ID of the clicked button - was set up inside settings window

// **timeExpired** -> true if time expired, means that the reward can be granted

private void RewardButtonClicked(TimerButtonIDs buttonID, bool timeExpired)

```
{
    if(timeExpired)
    {
        //if(buttonID == RewardButtonIDs.YourButtonID)
        //{
            //give reward for this button ID
        //}
    }
    else
    {
        //not ready yet, you have to wait
    }
}
```




3. USER GUIDE - TIMER BUTTON

Get the remaining time for a specific button

- **GleyDailyRewards.TimerButton.GetRemainingTime(TimerButtonIDs.ButtonID);**

Reset timer for a specific button

- **GleyDailyRewards.TimerButton.ResetTimer(TimerButtonIDs.ButtonID);**

Add the amount of time to the specified button

- **GleyDailyRewards.TimerButton.AddTime(TimerButtonIDs.ButtonID, new TimeSpan(0, 1, 0));**

Remove the amount of time from the specified button

- **GleyDailyRewards.TimerButton.RemoveTime(TimerButtonIDs.ButtonID, new TimeSpan(0, 1, 0));**



4. USER GUIDE - CALENDAR POPUP

Show the calendar

- **GleyDailyRewards.Calendar.Show();**

Subscribe to click listener from any script so you can grant and store the reward

- **GleyDailyRewards.Calendar.AddClickListener(CalendarButtonClicked);**

```
// dayNumber -> current clicked day  
// rewardSprite -> the sprite of the reward  
// rewardValue -> the reward value for current day
```

The screenshot shows a Unity Inspector window for a GameObject named "Day 1". It has two public variables: "Texture" and "Value". The "Texture" field is set to "DR_CoinPlate" with a small icon to its left and a dropdown arrow to its right. The "Value" field is set to "10". At the bottom of the Inspector, there is a button labeled "Remove Day".

```
private void CalendarButtonClicked(int dayNumber, int rewardValue, Sprite rewardSprite)  
{  
    yourRewardVariable += rewardValue;  
}
```

Get the remaining time for next day

- **GleyDailyRewards.Calendar.GetRemainingTimeSpan();**



5. PLAYMAKER SUPPORT

- **Supported Playmaker Actions:**
 - **Open Calendar**
 - **Timer Button Clicked**
 - **Get Remaining Time**
 -
- **Timer Button Clicked details**

The screenshot displays the Unity Playmaker interface. At the top, there are two panels showing the 'Variables' tab for the 'Timer Button Clicked' action. The left panel shows the 'Variables' tab with a list of variables: 'Event' (Used 2), 'FINISHED' (Used 2), 'ButtonClicked' (Used 2), and 'TimerActive' (Used 2). The right panel shows the 'Variables' tab with a list of variables: 'Name' (Used 3), 'ButtonID' (Used 2, Type: Enum), and 'RemainingTime' (Used 2, Type: String).

Below these panels is the 'ScriptsHolder : FSM' window, which shows a state machine diagram. The diagram starts with a 'START' state, which transitions to a 'ListenForClicks' state. The 'ListenForClicks' state has two outgoing transitions: one to a 'Click' state (labeled 'ButtonClicked') and one to a 'TimerActive' state (labeled 'TimerActive'). Both the 'Click' and 'TimerActive' states have a 'FINISHED' label, indicating they lead back to the 'ListenForClicks' state.

On the right side of the 'ScriptsHolder : FSM' window, the 'Timer Button Clicked' action is configured. The configuration includes the following fields:

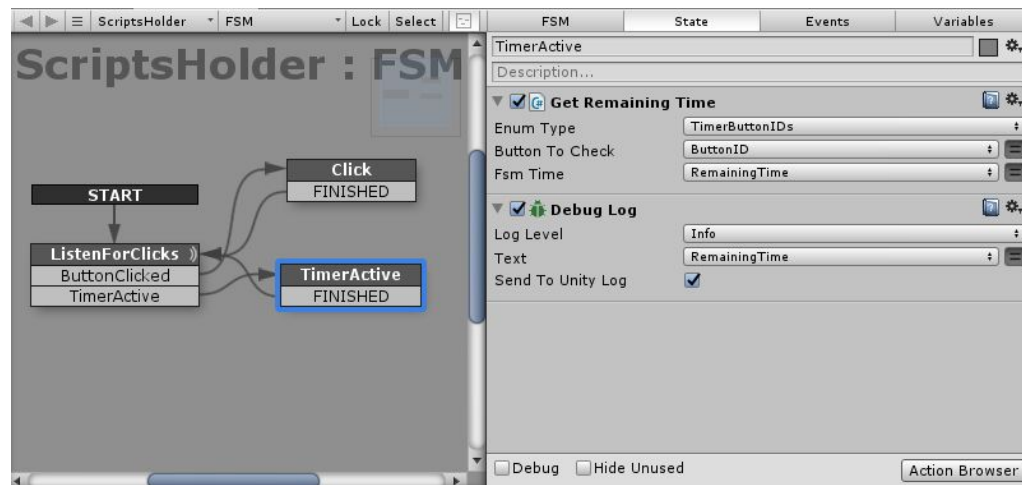
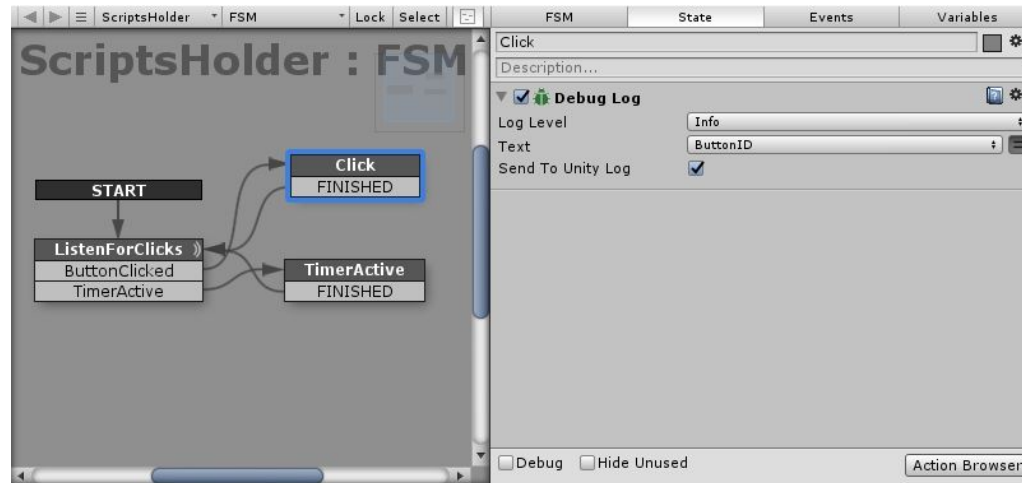
- Event Target: Self
- Button Clicked: ButtonClicked
- Timer Active: TimerActive
- Enum Type: TimerButtonIDs
- Clicked Button ID: ButtonID

At the bottom of the 'ScriptsHolder : FSM' window, there are checkboxes for 'Debug' and 'Hide Unused', and an 'Action Browser' button.



5. PLAYMAKER SUPPORT

- **Timer Button Clicked details**





5. PLAYMAKER SUPPORT

- Open Calendar details

The screenshot displays the Unity PlayMaker interface. The top panel shows the 'Events' tab with two events: 'CalendarButtonClicked' and 'Clicked', both marked as 'Used' with a count of 2. The bottom panel shows the 'Variables' tab with three variables: 'dayNumber' (Int), 'rewardSprite' (Sprite), and 'rewardValue' (Int), all marked as 'Used' with a count of 2.

The main window shows the 'ShowCalendar : FSM' state machine. The flowchart includes the following states and transitions:

- START** (black box) transitions to **ClickListener** (blue box).
- ClickListener** (blue box) has a 'Clicked' event and transitions to **OpenCalendar** (grey box).
- OpenCalendar** (grey box) has a 'CalendarButtonClicked' event and transitions to **ClickedDay** (black box).
- ClickedDay** (black box) transitions back to **ClickListener** (blue box).

The right panel shows the configuration for the 'ClickListener' state:

- UI Button On Click Event** (checked)
- Game Object**: Use Owner
- Event Target**: Self
- Send Event**: Clicked

At the bottom, there are checkboxes for 'Debug' and 'Hide Unused', and an 'Action Browser' button.



5. PLAYMAKER SUPPORT

- Open Calendar details

The screenshot shows the Unity Playmaker FSM editor for a state named 'OpenCalendar'. The left pane displays the state machine diagram with the following flow: START → ClickListener (Clicked) → OpenCalendar (CalendarButtonClicked) → ClickedDay. The right pane shows the configuration for the 'OpenCalendar' state:

Event Target	Value
Button Clicked	CalendarButtonClicked
Day Number	dayNumber
Reward Value	rewardValue
Reward Sprite	rewardSprite

At the bottom of the right pane, there are checkboxes for 'Debug' and 'Hide Unused', and an 'Action Browser' button.

The screenshot shows the Unity Playmaker FSM editor for a state named 'ClickedDay'. The left pane displays the state machine diagram with the following flow: START → ClickListener (Clicked) → OpenCalendar (CalendarButtonClicked) → ClickedDay. The right pane shows the configuration for the 'ClickedDay' state:

Log Level	Text	Send To Unity Log
Info	rewardSprite	<input checked="" type="checkbox"/>
Info	rewardValue	<input checked="" type="checkbox"/>
Info	dayNumber	<input checked="" type="checkbox"/>

At the bottom of the right pane, there are checkboxes for 'Debug' and 'Hide Unused', and an 'Action Browser' button.

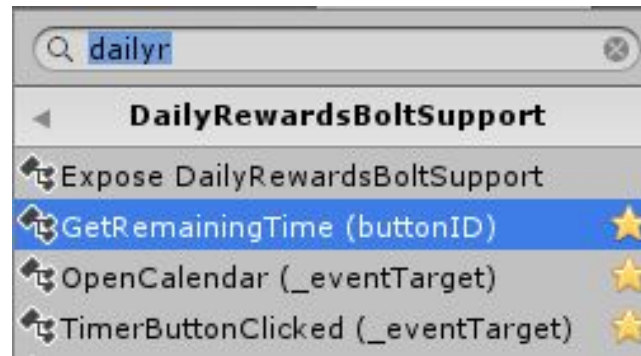


6. BOLT SUPPORT

- **Supported Bolt Actions:**
 - **Open Calendar**
 - **Timer Button Clicked**
 - **Get Remaining Time**

The above Bolt actions behavior is equivalent with corresponding methods from Section 3 - User Guide.

The above Bolt actions are located inside **DailyRewardsBoltSupport** script as shown below:



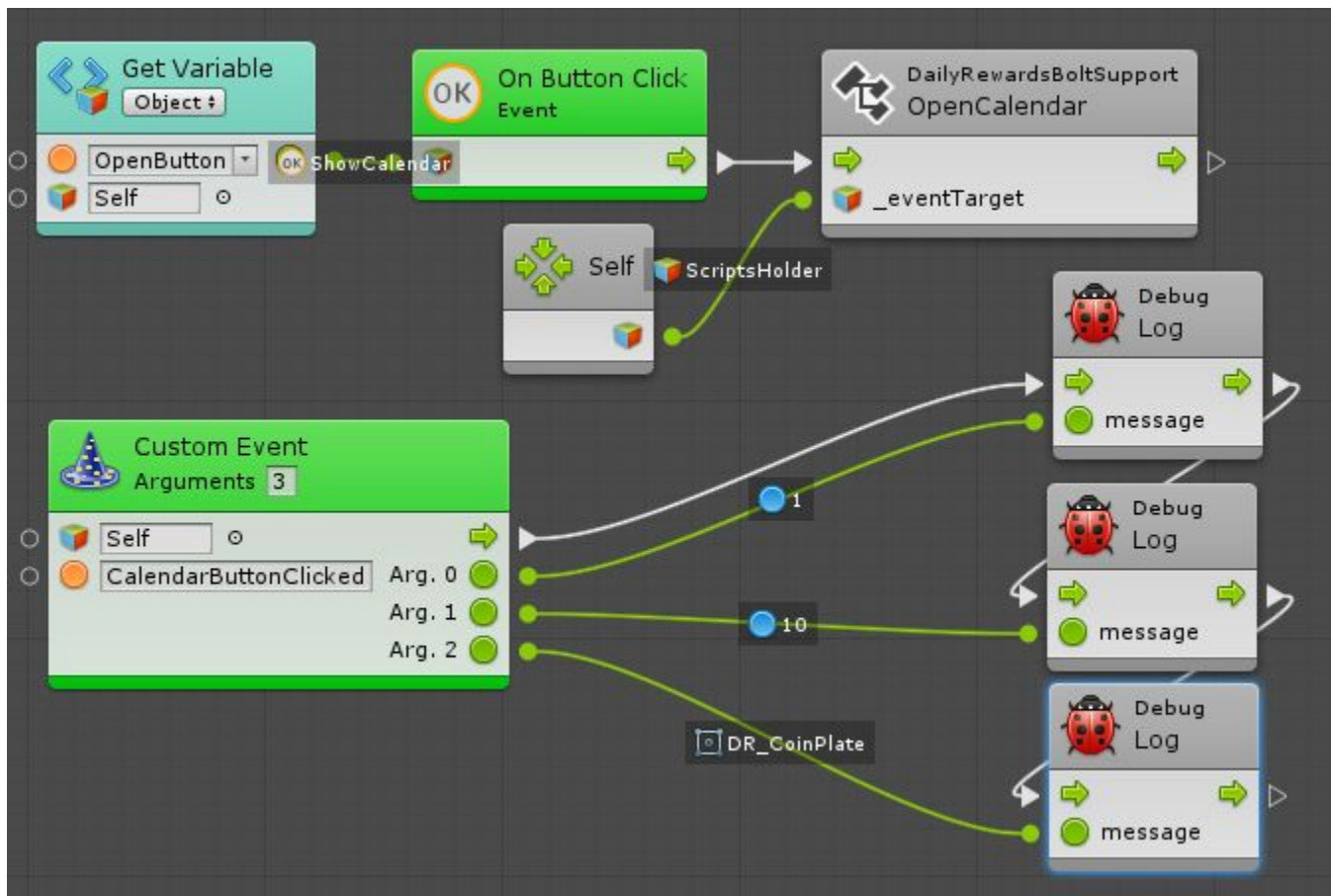


6. BOLT SUPPORT

- **Open Calendar details**

Custom event arguments:

Arg. 0 -> clicked day number; Arg. 1 -> clicked day reward; Arg. 2 -> clicked day sprite





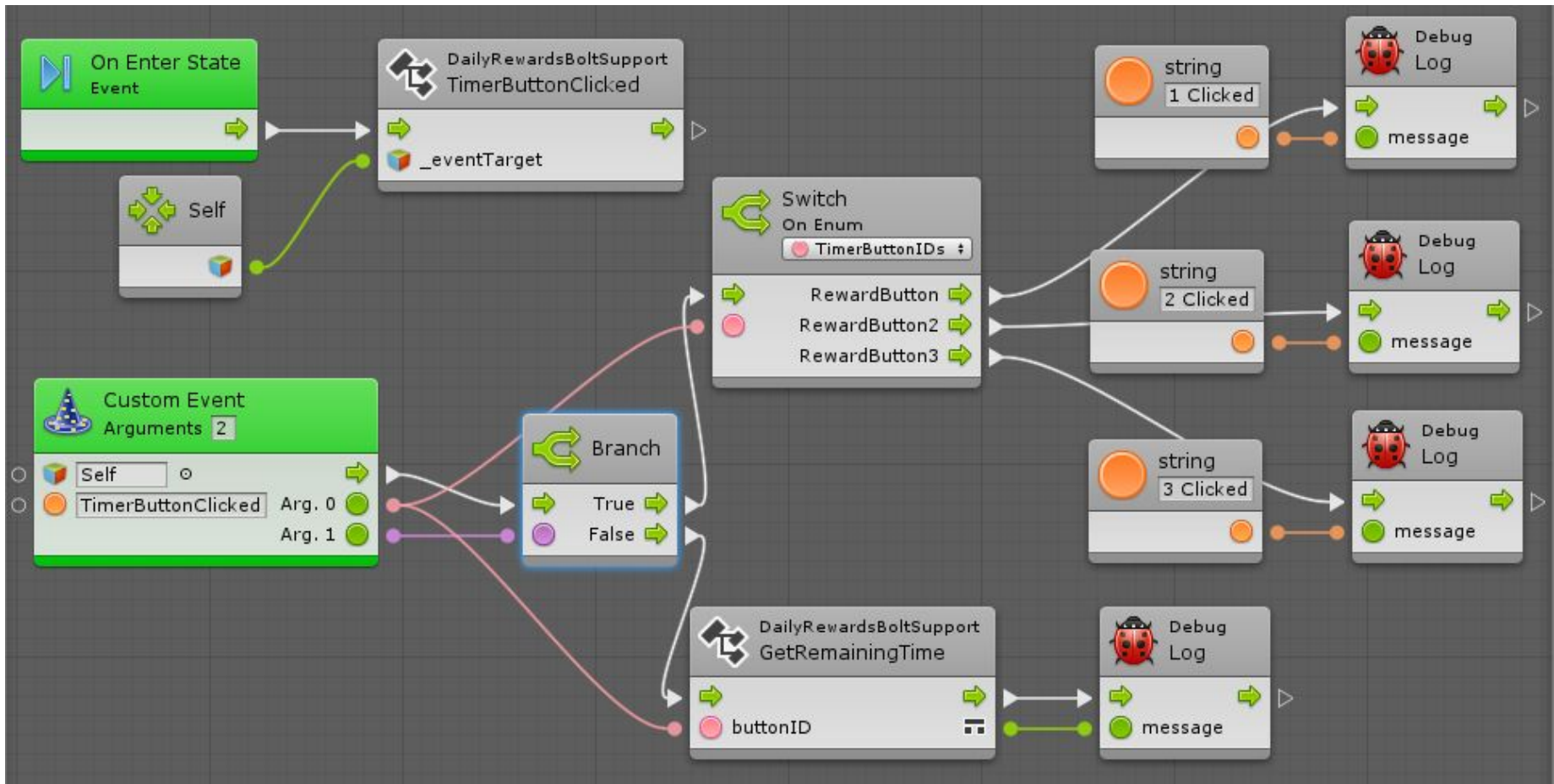
6. BOLT SUPPORT

- **Timer Button details**

Custom event arguments:

Arg. 0 -> TimerButtonID enum - represents the clicked button;

Arg. 1 -> bool - time expired





7. EXAMPLE 1 - Timer Buttons

You can find the example test scene here:

Assets/GleyPlugins/DailyRewards/Example/TimerButtonExample.unity

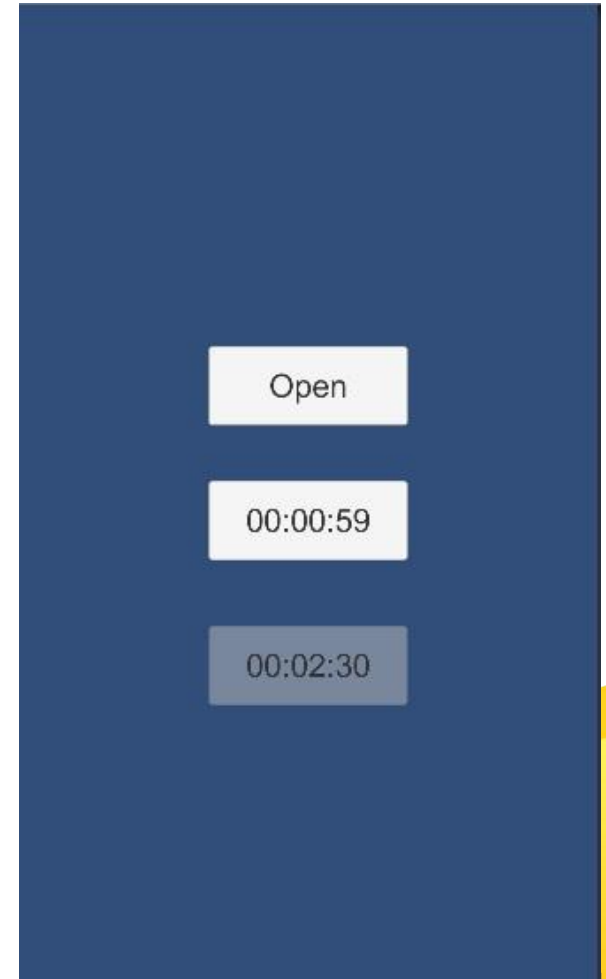
1. Setup all timer buttons from Settings Window:

TIMER BUTTON SETUP:

Button ID (Unique)	RewardButton		
Time to pass:	h: 0	m: 1	s: 0
Unlocked at start	<input checked="" type="checkbox"/>		
Text after complete	Open		
Interactable when unavailable	<input type="checkbox"/>		
Remove Timer Button			

Button ID (Unique)	RewardButton2		
Time to pass:	h: 0	m: 2	s: 0
Unlocked at start	<input checked="" type="checkbox"/>		
Text after complete	Open		
Interactable when unavailable	<input checked="" type="checkbox"/>		
Remove Timer Button			

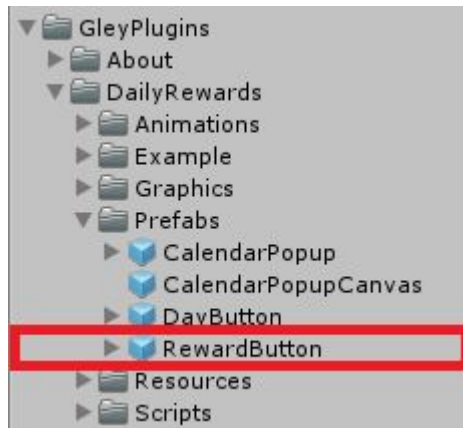
Button ID (Unique)	RewardButton3		
Time to pass:	h: 0	m: 3	s: 30
Unlocked at start	<input type="checkbox"/>		
Text after complete	Open		
Interactable when unavailable	<input type="checkbox"/>		
Remove Timer Button			



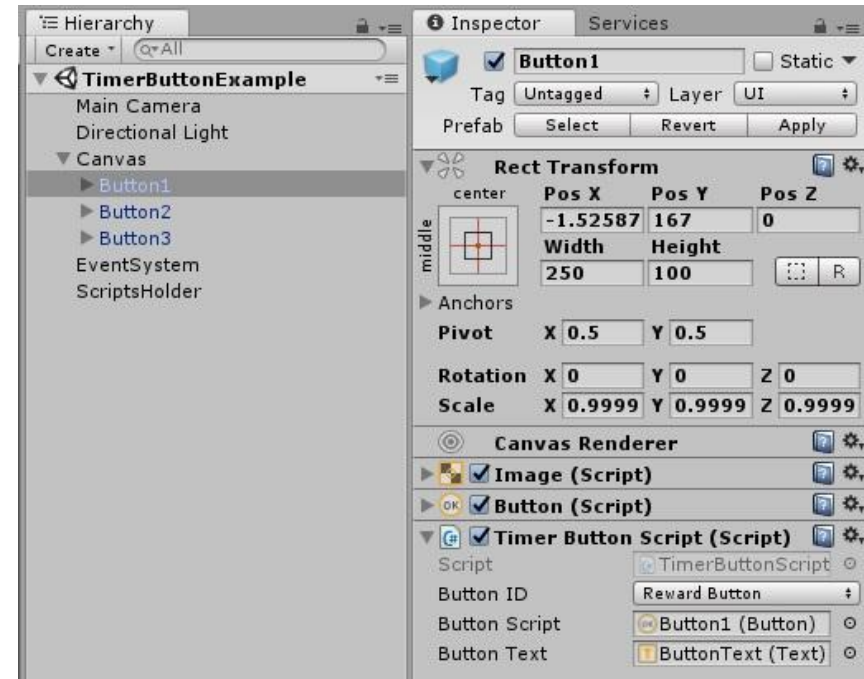
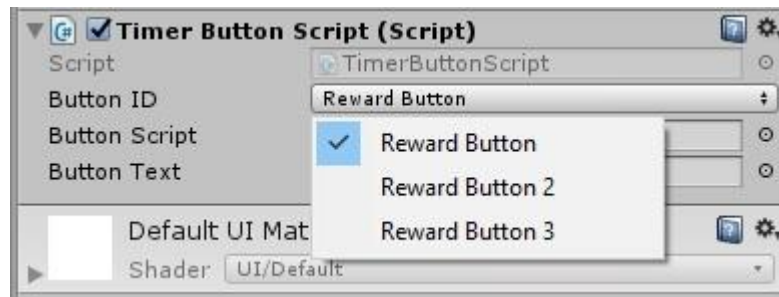


7. EXAMPLE 1 - Timer Buttons

2. Drag reward button prefab to the scene:



3. Associate the button with your settings by selecting the corresponding button ID from button script:





8. EXAMPLE 2 - Calendar

You can find the example test scene here:

Assets/GleyPlugins/DailyRewards/Example/CalendarExample.unity





8. EXAMPLE 2 - Calendar

Calendar customization:

1. Create your rewards in settings window

CALENDAR SETUP:

Calendar Prefab

Calendar Canvas

First day unlocked ☐

Restart at end of days ☒

Time to pass: h: m: s:

Day 1

Texture

Value

Day 2

Texture

Value

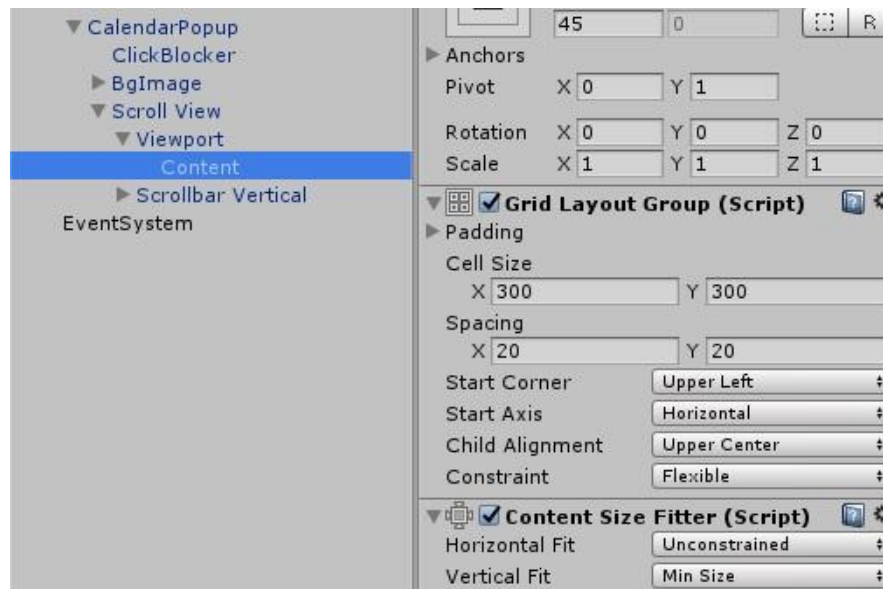


8. EXAMPLE 2 - Calendar

Calendar customization:

2. Modify day size:

- Go to CalendarPopup prefab -> Scroll View -> Content and change the cell size and spacing



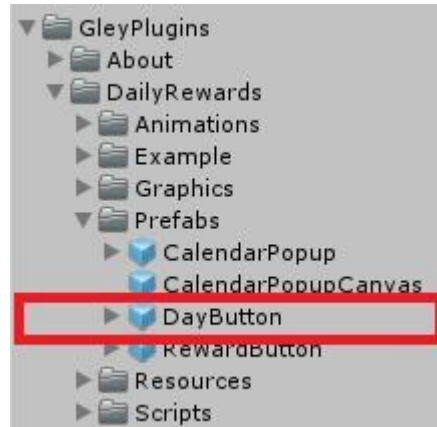


8. EXAMPLE 2 - Calendar

Calendar customization:

2. Modify day button appearance:

- Open prefab from:



- Change the appearance
- Sprites are change with the state of the button as follows:
- **Claimed sprite** - active for past days
- **Current sprite** - active for current day but timer is active(cannot be claimed)
- **Available sprite** - active for current day (can be claimed)
- **Locked Sprite** - active for future days

