

# Event Editor

SUMMARY .....	2
COMPATIBILITY .....	2
CODE ACCESSIBILITY .....	2
WORKFLOW WITH GAMEEVENT .....	3
GAMEEVENT AND PAGES .....	3
WORKFLOW WITH LOCAL AND GLOBAL SWITCHES .....	3
SERIALIZATION OF LOCAL AND GLOBAL SWITCHES .....	4
SETUP A NEW GLOBAL SWITCH CATALOG .....	5
SETUP OF THE EVENT EDITOR .....	6
SETUP OF THE PAGE EDITOR .....	7
SETUP OF PAGE CONDITIONS .....	8
COLOR CODE FOR PAGES .....	9
COLOR CODE FOR CONDITIONS .....	9
LOCAL SWITCH WINDOW .....	10
GLOBAL SWITCH WINDOW .....	11

## Summary

The Event Editor is a tool specifically created to help the workflow with GameObjects in Unity. It is designed for projects that need a similar handling of GameObject inside a scene.

## Compatibility

The Event Editor is written in .NET version 4.0 standards and is tested in Unity 2018.3 and upwards.

## Code Accessibility

Most of the features of the Event Editor and their components can be controlled by via code. All that has to be done is to use the using directive.

```
using LGP.EventEditor;
```

For more information about the GameEvent functions check out the source code or carefully read this documentation. All relevant Methods are mentioned in the documentation and commented in the source code.

## Workflow with GameEvent

This tool features a new component named GameEvent with its own editor. This component act now as a director for the GameObjects and takes care of conditions and function processing

⇒ This will be elaborated more in their respecting chapters.

Since the GameEvent takes care of directing the GameObject, the user still has to implement functionality. With the help of the Event Editor the user can then bind the functionality of the GameObject and its many other components to the conditions of the GameEvent.

This saves a lot of work and streamlines workflow with a big GameObject count. In Addition, it gives direct overview and clarity inside the inspector, how the functions interact with the GameObject.

GameEvents are only active if their GameObject is active inside an active Scene. If the state changes the GameEvent will reset itself. All active GameEvents are stored inside a static list, which can be accessed by code.

```
GameEvent.GameEvents : List<GameEvent>
```

## GameEvent and Pages

A GameEvent holds a list of Pages. A page represents a potential state the GameEvent can become. A page has a list of conditions, which determine if the page may become active or not.

The key concept of the Event Editor is the combination of these Pages and their Conditions. The GameEvent can then direct the logic flow during runtime, adapt new states and process user made functionality for this specific state.

- ! The GameEvent sequentially checks which page has its conditions positive first in the list. Therefore, the order of the list is an important factor. A page is considered active, if the first page's condition in the list all return positive.

When a page becomes active a special delegate `OnPageActive(EEPage page)` will be invoked. This enables the user to subscribe their functions to preemptively setup the GameObject to the active page. For instance, changing the RuntimeAnimator for a specific active page.

## Workflow with Local and Global Switches

The Event Editor works with two additional systems. Local and Global Switches. The main difference between these two Switches is the object they are stored in

- The Local Switch Catalog is stored on the GameEvent itself.
- The Global Switch Catalog is stored inside a scriptable object asset in the asset Menu.

This extends the conditional aspect of the GameEvent. Normally these conditions have to be implemented by code for each GameObjects. This tool however streamlines this approach and enables easier and clearer use of conditions inside the game.

- ! **The Global Switch Catalog is best stored separate inside a player save file.** The Global Switch asset in the asset menu only acts as an initial state of the Global Switches for the player when they start a new game with an empty save file.
- ! **Changes done to the global Switch Catalog will be hard saved regardless of being in play mode or not.** Please use the Global Switches Catalog in the asset menu as an initializer for a new save file.

## Serialization of Local and Global Switches

Local and Global Switches are both stored inside custom made serialized Dictionaries. These directories are then either stored on the GameEvent or Scriptable Object. Both objects enable access via code and editor to the user.

**A GameEvent doesn't store changes of its Local Switches, when the scene gets unloaded or the player quits the game.** Local Switches are only temporal existing inside the scene of the GameEvent and act only as conditional helpers for the user to setup the GameEvent.

To achieve that the GameEvent holds an internal Dictionary that stores the Initial States of the local Switches. Local Switches use this initial Catalog to reset themselves if needed. The Initial Catalog is accessible in the Editor but not in code.

- ! It is advised not to change Initial Local Switches during runtime without concrete concept. It has the potential to permanently change the GameEvent condition processing, resulting in a different logic flow.

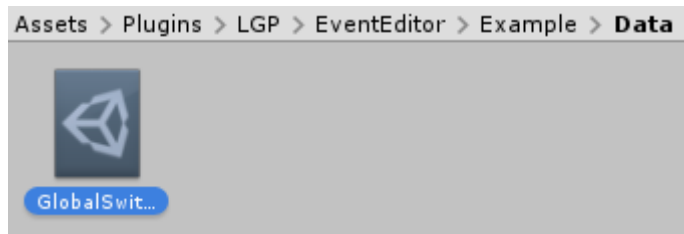
In certain cases, where changing initial Local Switch values proves to be useful, it is recommended to check the implementation in the source code.

## Setup a new Global Switch Catalog

To create a new Global Switch Catalog:

1. Navigate to the asset window or click on *Assets* on the menu bar.
2. Go to *Create* and click on *Global Switch*

A new Asset will spawn in the asset directory. This is the new Global Switch Catalog.

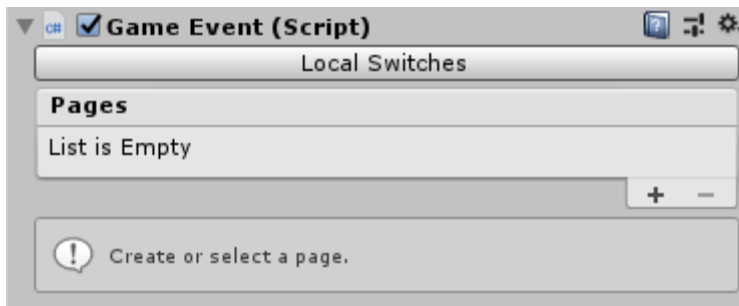


## Setup of the Event Editor

The Event Editor is the custom inspector for the GameEvent component.

1. Open up a scene.
2. Create a new GameObject in the Scene.
3. Add the component “GameEvent” to the GameObject via the *Component* tab on the menu bar or via *Add Component* in the GameObject’s inspector.

The inspector has now an additional component. This is the Event Editor for the GameEvent.



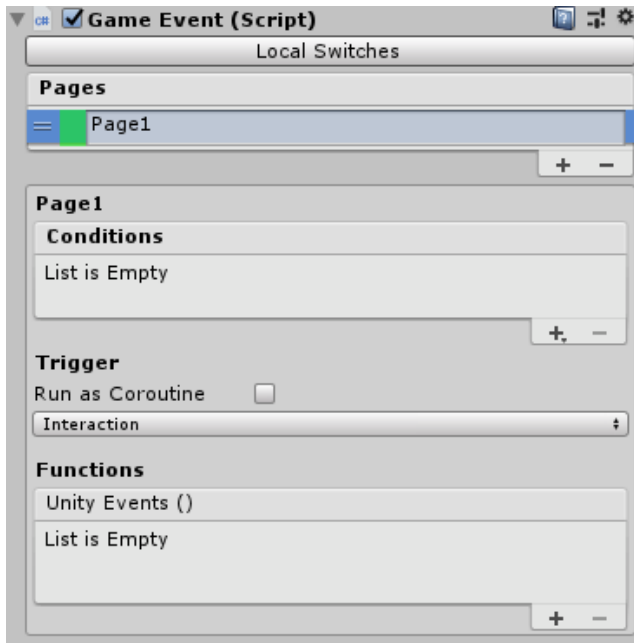
- **Local Switches**  
Clicking this Button opens the Local Switch Window of this GameEvent.  
⇒ How to setup Local Switches go to the chapter about the Local Switch Window.
- **Pages**  
This is the list of pages, which are stored on this GameEvent. Pages can be added, moved, selected or deleted from the Event Editor. The order in which these pages are setup is relevant for the page active check.  
⇒ How to setup Pages go to the chapter about the Page Editor.

## Setup of the Page Editor

To create a new Page inside the Event Editor:

1. Go to the GameEvent's Event Editor
2. Press on the **Pages** list the "+" symbol.

The **Pages** list now has a new page added to it. The Event Editor automatically selects the new Page and extends its inspector with new Elements.



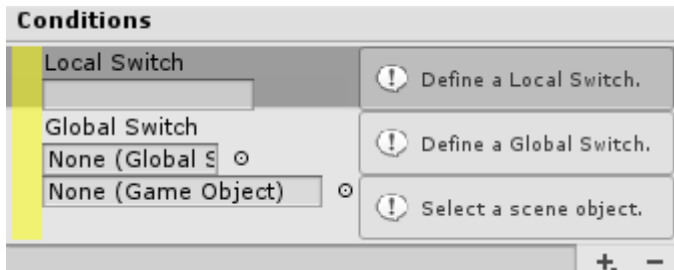
- **Page 1**  
This is the Name of the Page. It can be set manually by the user. The GameEvent doesn't care what pages are called, it is purely for user clarity.
  - ⇒ The conditions colors are more elaborated in the chapter about Color Codes.
- **Conditions**  
This is the condition list of the Page. Inside the user can add, remove and modify different types of conditions.
  - ⇒ The different kinds of conditions are explained in the chapter about conditions.
- **Trigger**  
The Trigger determines how the Functions below are processed.
  - Interaction:** The Functions are invoked when the `Interact()` Method is called of the GameEvent or the static variation `GameEvent.Interact(GameEvent gameEvent)`.
  - Autorun:** this trigger automatically triggers when the page becomes active.
  - Custom:** This trigger checks an interface method `IEECustomTrigger.CustomTrigger(): bool` if it returns `true`. This trigger can be implemented anywhere on the GameObject.
  - Run as Coroutine:** This toggle field determines if the functions are processed in a coroutine or not. Coroutines enables function invoke to run parallel to other running system in the game.
- **Functions**  
This is a list of Unity Events where GameObject can subscribe their functions to.

## Setup of Page Conditions

A Page can have zero or multiple conditions. To create a new Condition do:

1. Press “+” on the **Conditions** list.
2. Choose the type of the condition.

Now the **condition** list has a new condition. Depending on the type of the condition, the list displays different results.



- **Local Switch Condition**

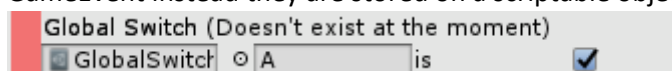
This condition check the local Switch directory of the GameEvent.



The User can define the key and value of conditional switch, even if corresponding Local Switch doesn't exit at the moment. The inspector will always inform the user about the availability of the Local Switch.

- **Global Switch Condition**

This works similarly to the Local Switch. However, Global Switches are not stored on the GameEvent instead they are stored on a scriptable object of type GlobalSwitch.



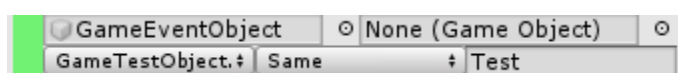
The user can define the Global Switch Object, the key and value of the conditional global Switch. The inspector will tell about the availability of the Global Switch.

- **Game Object**

This condition takes a scene Object as target. The user can then select a Field, Property or Method from the selected GameObject's components.

There are however certain conditions to keep in mind

- ! Only Fields, Properties and Methods with the attribute `[Conditional]` tagged on the source code will be exposed to the Event Editor.
- ! Only Fields, Properties and Method with return type of either: `bool`, `int`, `float` or `string` are exposed to the Event Editor.
- ! Only Methods without any parameters are exposed to the Event Editor.



The user can then select the conditional operator and the target value of the condition. Also, this condition lets the user compare values of two GameObjects in the scene. All that is to do is to define both GameObjects and select their Field, Property or Method. The inspector will automatically filter the other GameObjects list to the corresponding type.



## Color Code for Pages

	The Page's condition is met and are can be potentially become active if first in list.
	The Page's condition is not met and probably won't become active during runtime.
	Marks the currently active Page. (Only during runtime in play mode!)

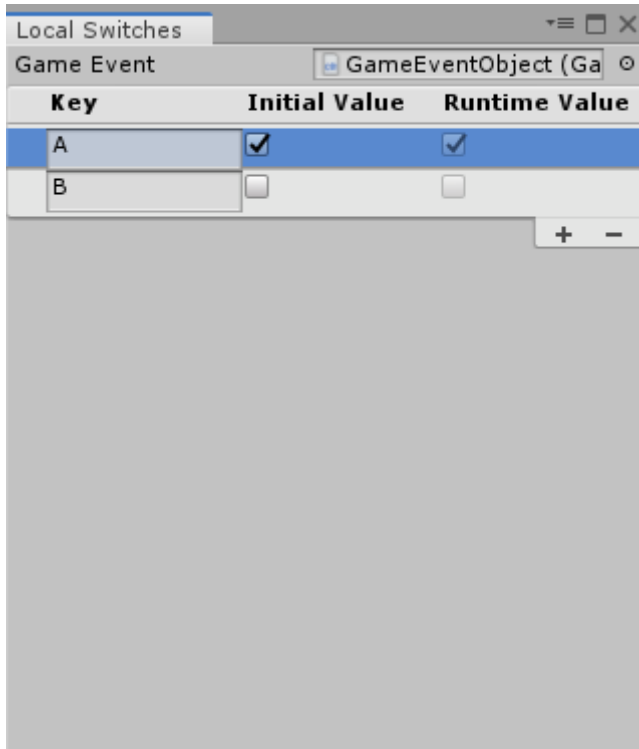
## Color Code for Conditions

	This Condition will return positive.
	This Condition will return negative.
	Empty Condition. This condition will not influence the condition outcome. It marks empty conditions for clarity reasons.

## Local Switch Window

This window can be accessed by clicking on *Tools => Event Editor => Local Switch* or pressing the *Local Switches* button found on the Event Editor.

The Window features a list of all Local Switches stored on the Game Event.



- **Game Event**  
This is the target GameEvent of this window to derive the Local Switches from. Depending on where this window was accessed from, the target has to be set up manually by the user. Just select the GameEvent in the scene.
- **The List**  
The Local Switch list can Add, modify or remove Global Switches.  
**Key**  
This is the Key of the Local Switch. The name has to be **unique**.
- **Initial Value**  
This is the value setup in the Event Editor. The GameEvent will reset its Local Switch Catalog to these values.
- **Runtime Value**  
This is the actual value of the Local Switch, which will be used for condition checking.

These changes can only be modified in Editor Mode and will be disabled if the game is in play mode.

All changes done are then reflected by the GameEvent object.

## Global Switch Window

This window can be accessed by clicking on *Tools => Event Editor => Global Switch*.

The Window features a list of all Global Switches stored on Scriptable Object



- Global Switch  
This is the target Catalog of this window to derive the Global Switches from. Just select the Scriptable Object from the asset menu.
- List  
The Global Switch list can Add, modify or remove Global Switches.
- Key  
This is the Key of the Global Switch. The name has to be **unique**.
- Value  
This is the editor and runtime value of the global Switch.

These changes can only be modified in Editor Mode and will be disabled if the game is in play mode.

All changes done are automatically reflected by the GameEvent object.