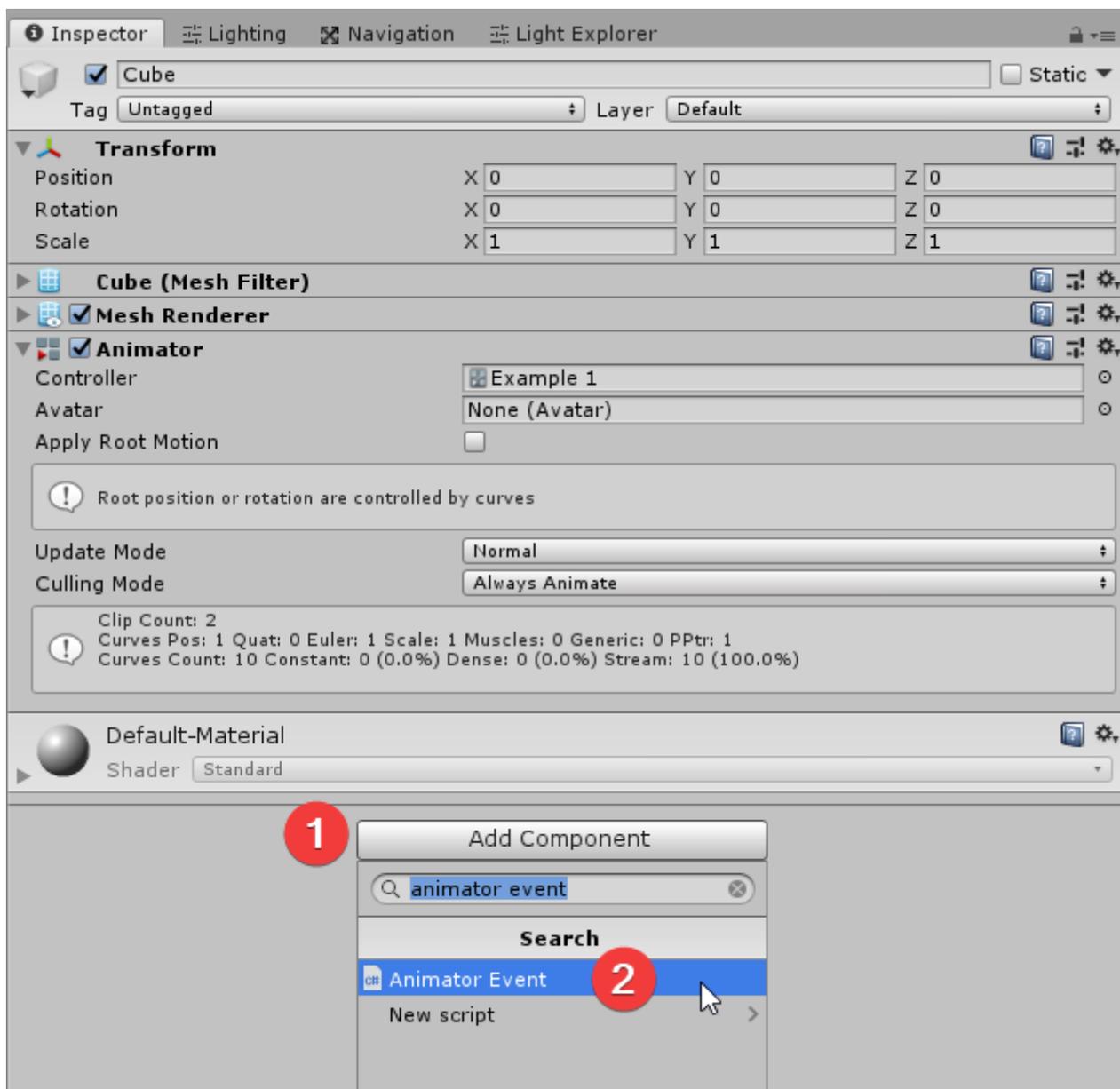


Animator Events

Support by email (found in <https://ashkatchap.nukefist.com/>) or through Discord <https://discord.gg/WyxYWPCG7F>

How to use

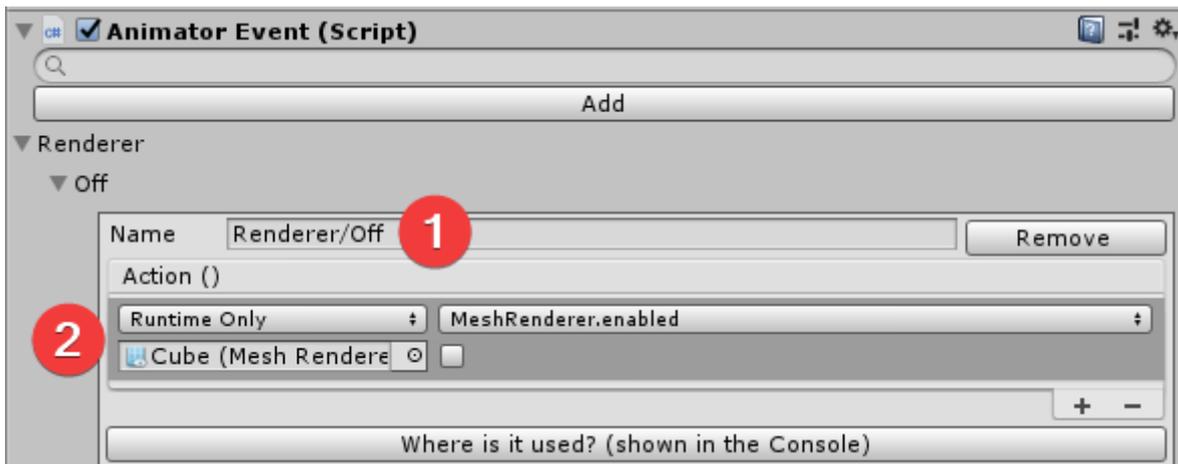
While inspecting an Animator, add the component **Animator Event**. (there can be more than one attached to the same GameObject).



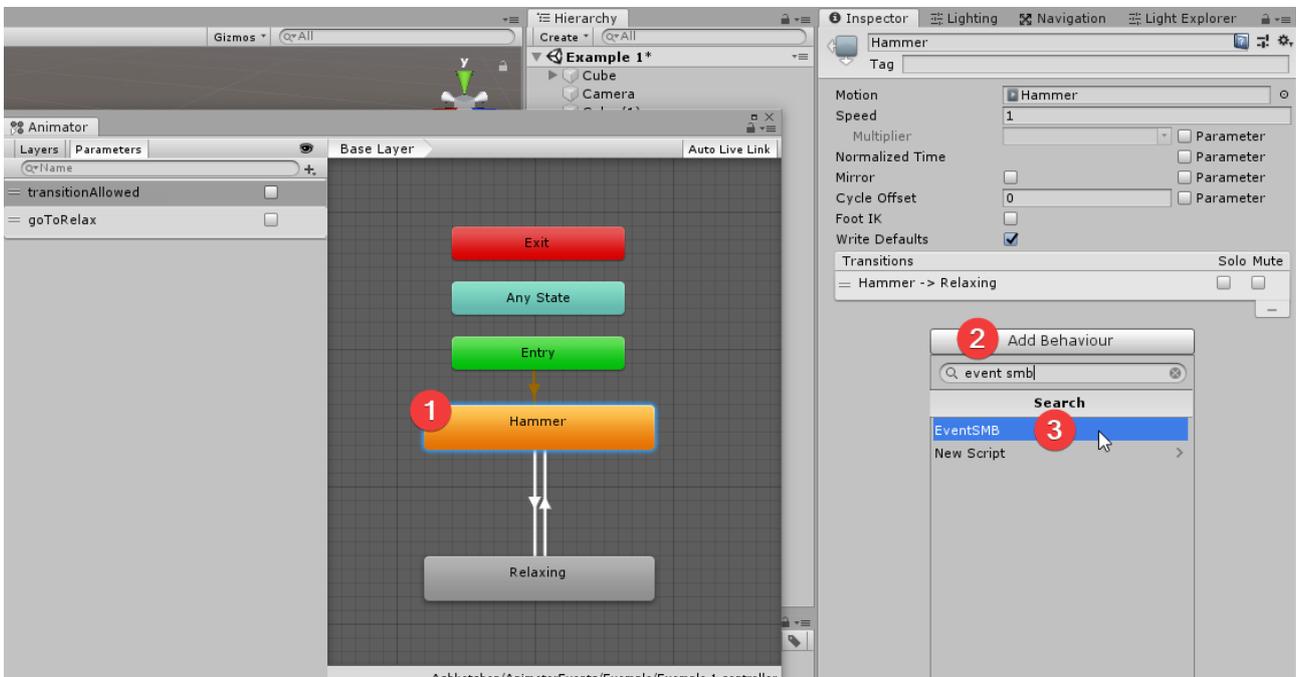
Add events to the **Animator Event**. They can be added, removed and renamed at any time.



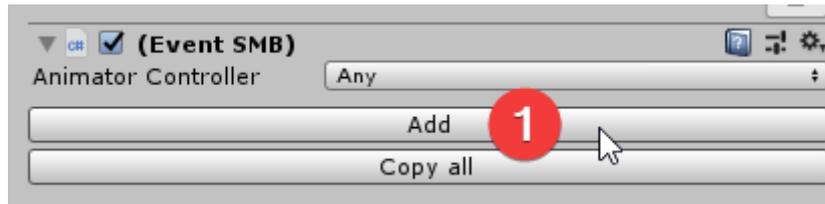
Give the event a name. Use the character / to organize events in "folders". Then add actions that will be executed when this event is called.



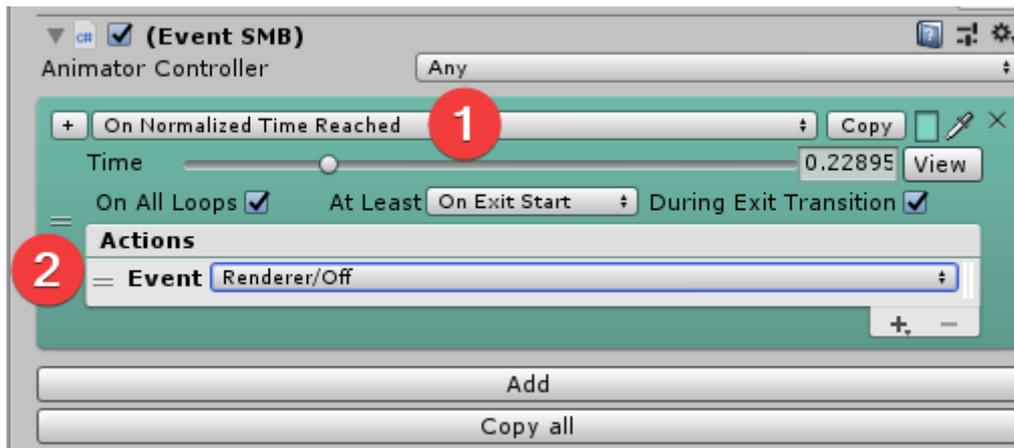
Inspect a state on the Animator and add to it the Component **Event SMB**.



Add an entry to **Event SMB**.

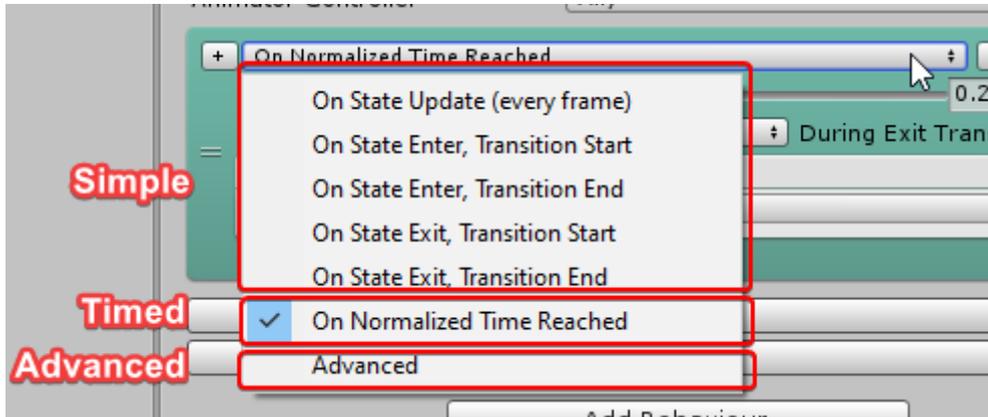


Setup the conditions for this entry, then specify which actions will be performed when the conditions are met.



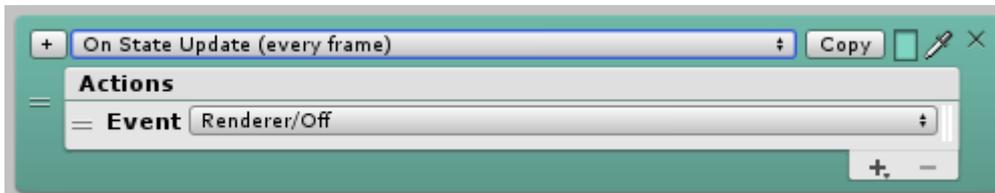
Event conditions

There are 3 types of conditions: Simple, Timed and Advanced



Simple conditions

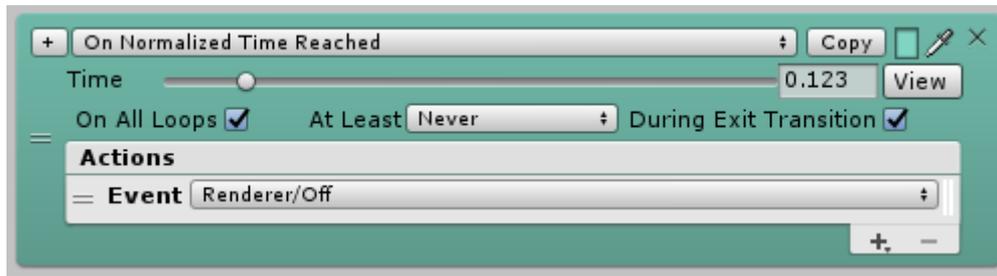
Don't have any type of extra configuration and do what they specify



- **On State Update**
This condition evaluates always to TRUE.
- **On State Enter, Transition Start**
When this state starts playing. If a transition is used, at the first frame the transition starts playing.
- **On State Enter, Transition End**
When this state starts playing. If a transition is used, at the frame the transition stops playing.
- **On State Exit, Transition Start**
When this state stops playing. If a transition is used, at the first frame the transition starts playing.
- **On State Exit, Transition End**
When this state stops playing. If a transition is used, at the frame the transition stops playing.

Timed conditions

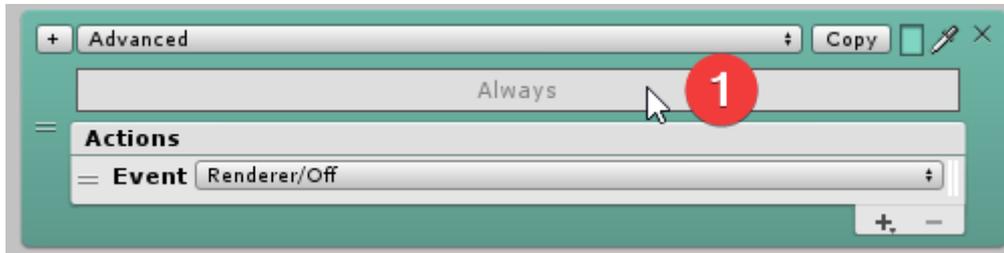
Have extra configuration settings to specify some common settings.



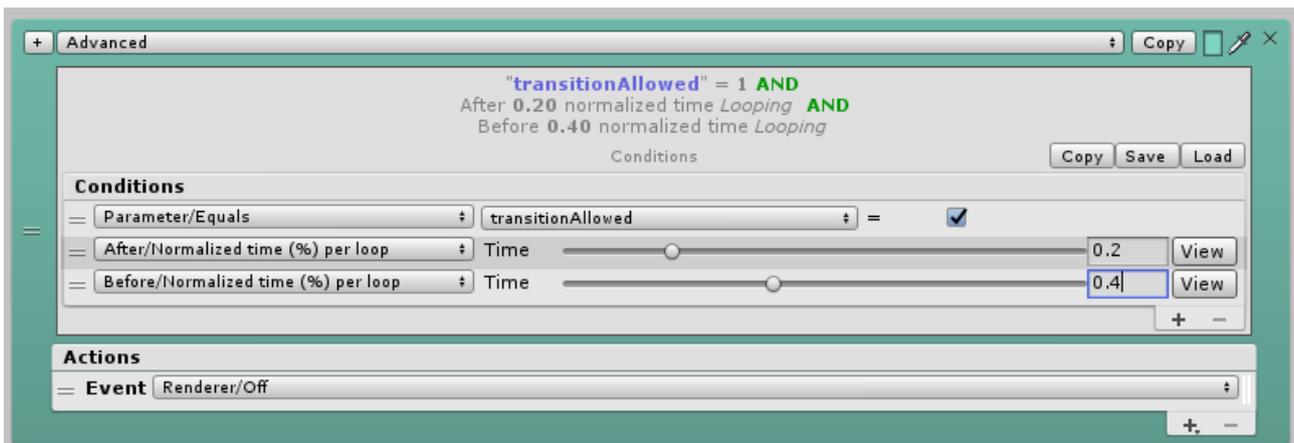
- **Time**
In normalized time, at which point in the animation this condition is TRUE. When scrolling the time or clicking on the View button, a preview of the animation at the selected point in time will be visible in the Scene.
- **On All Loops**
Execute the actions every time the state loops. Otherwise only the first loop will execute them. This is useful in case the state loops several times but only the first loop must execute the event.
- **At Least**
Execute the actions when exiting this state if it hasn't yet at least once.
 - **On Exit Start**
When the state starts exiting.
 - **On Exit End**
When the state finishes exiting.
 - **Never**
No extra logic will be used.
- **During Transition**
Allow executing actions while an exit transition is playing. If FALSE then events won't be executed during a transition even if the normalized time is reached.
This setting doesn't affect "At Least" from working, but if this is TRUE, "At Least" is "On Exit End" and the normalized time is reached during the exit transition, then the event will be executed during the transition and not on exit.

Advanced conditions

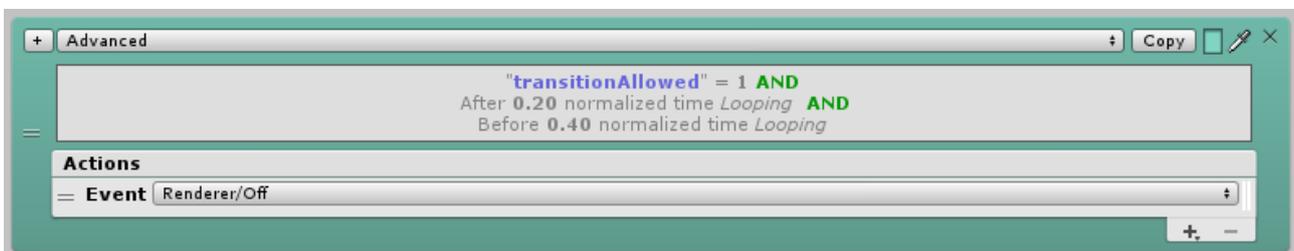
All other conditions are internally expressed as an advanced condition.
To edit them, click on the box.



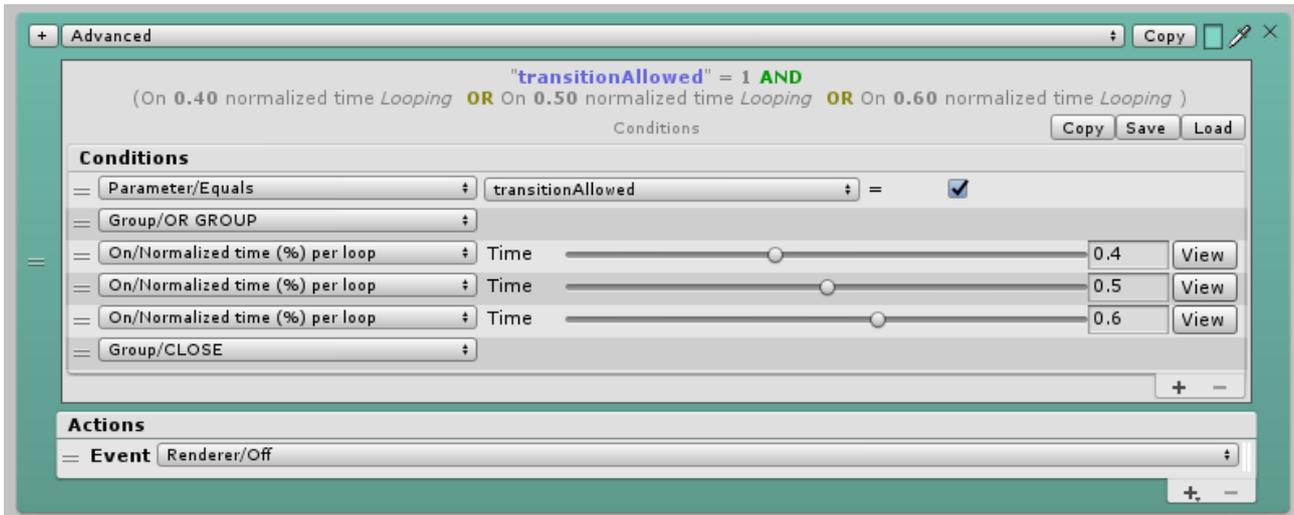
This example shows a condition that is true if the parameter “transitionAllowed” is true (a boolean parameter on the animator in this example) and the looped normalized time (that means, the logic is the same every time the animation loops) is in between 0.2 (20%) and 0.4 (40%).
This could be used to set a parameter to a certain value during a certain time frame of the animation, for example as a kind of window, then a transition could use that parameter as a condition.



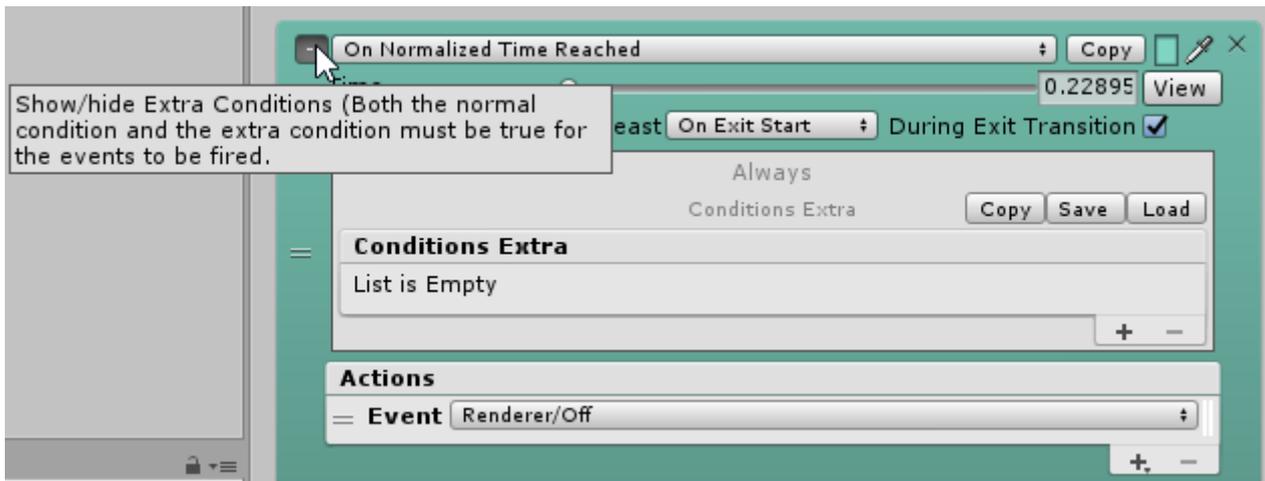
By clicking on the box, it gets collapsed again, showing only a plain text of the condition



This other example shows a condition that is only true if the parameter “transitionAllowed” is true and the looped normalized time of the state matches either 40%, 50% or 60% of the animation.



Click on the + button of an **Event SMB** entry to add extra conditions that must also be met to execute the actions. This adds an extra “Advanced Event” type of condition group.



There are 2 types of conditions: Group and the rest.

Group (Group a series of conditions).

- **AND**: Legacy. Please use **AND GROUP** as it's easier to understand.
- **OR**: Legacy. Please use **AND GROUP** as it's easier to understand.
- **NOT**: Negate the contents of this group.
- **CLOSE**: Puts end to a group. Isn't needed if the end is the last condition.
- **AND GROUP**: Between this and CLOSE, ALL conditions MUST be TRUE.
- **OR GROUP**: Between this and CLOSE, AT LEAST ONE condition MUST be TRUE.

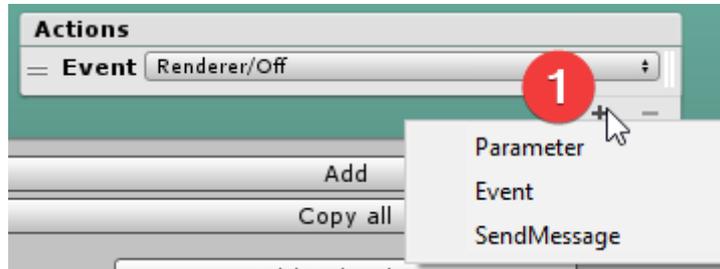
The rest (simple, each one evaluates to TRUE or FALSE).

Parameter/...	Check the value of a parameter.
Layer Index/...	Check the value of the current layer index.
Layer Weight/...	Check the value of the current layer weight.
On/Normalized Time	Once when a certain normalized time is reached.
On/Normalized Time Per Loop	Once per loop when a certain normalized time is reached.
On/Exit Transition Ends	When exiting the state, after the transition ends. TRUE the last frame the state is active.
After/Normalized Time	After a certain normalized time is reached.
After/Fixed Time (seconds)	After a certain number of seconds is reached.
After/Normalized Time Start transition	After a certain normalized time of the start transition is reached.
After/Normalized Time Exit transition	After a certain normalized time of the exit transition is reached.
After/Enter transition ends	After the entering transition ends.
After/Exit transition starts	After starting exiting the state.
Max N Times/After Start	This entry has been executed at most N times since entered the state.
Max N Times/Per Loop	This entry has been executed at most N times since last loop.
Delegate Condition	Dynamically generated delegates through code during runtime that return TRUE/FALSE.
Before/...	The contrary as After.

(Notice: After/... evaluates to TRUE also if we are on the same value. E.g. "After/Fixed Time" 3.5 will be to TRUE at exactly 3.5 and after that).

Event actions

There are 3 types of actions that can be performed by an entry if its condition evaluates to TRUE.



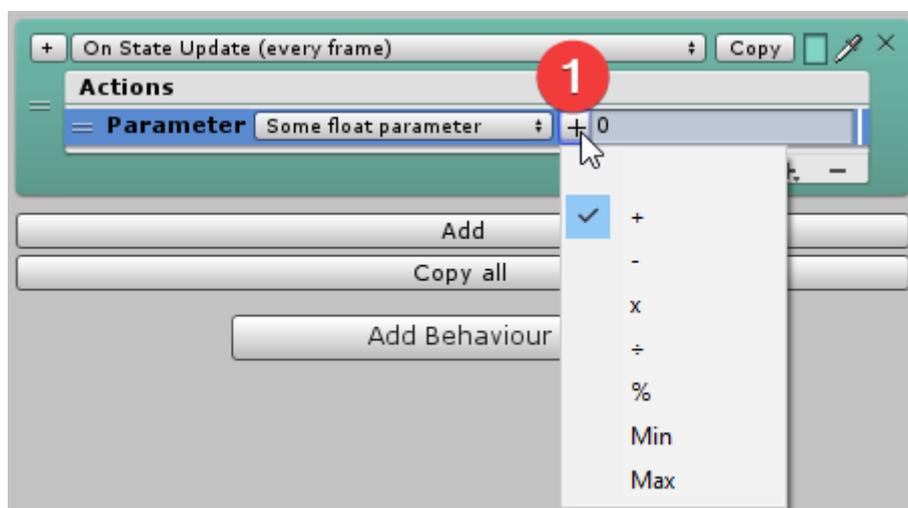
Parameter

Set the value of a parameter. There's a text field where we can input a value. We will call this value "NEW" to make examples easier.



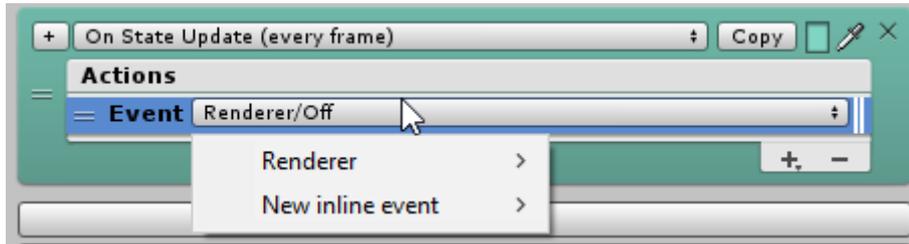
Perform a simple math operation on the parameter. The final value of the parameter will be equal to:

- **[+]** Parameter plus NEW.
- **[-]** Parameter minus NEW.
- **[x]** Parameter times NEW.
- **[/]** Parameter by NEW.
- **[%]** Parameter modulo NEW.
- **[Min]** Minimum value between parameter and NEW.
- **[Max]** Maximum value between parameter and NEW.



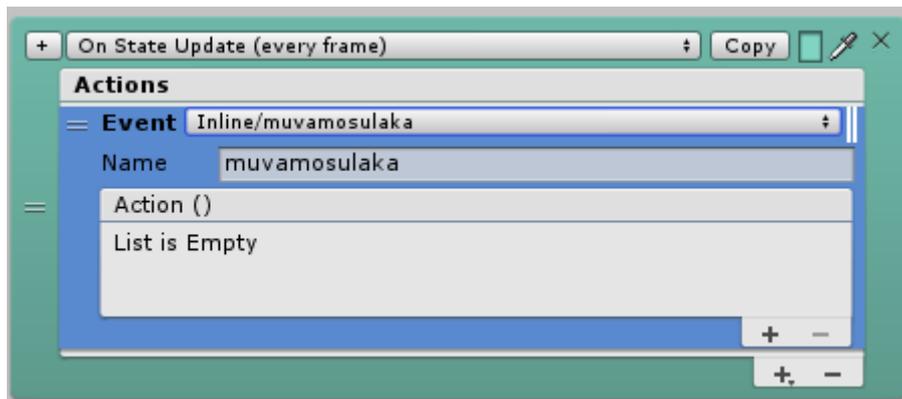
Event

Execute an event. You can either choose an existing one or to create a new one “New inline event”.



To be able to view the list of events, you need to before click on the GameObject that has the **Animator Event** component.

Inline events are stored inside the Animator Event, like any other event, but it will not show in the list of events you can choose in the dropdown. After creating one, a new random name will be given to it that you can change. You can promote an inline event to a normal event and vice versa by having its name start or not with “Inline/”.

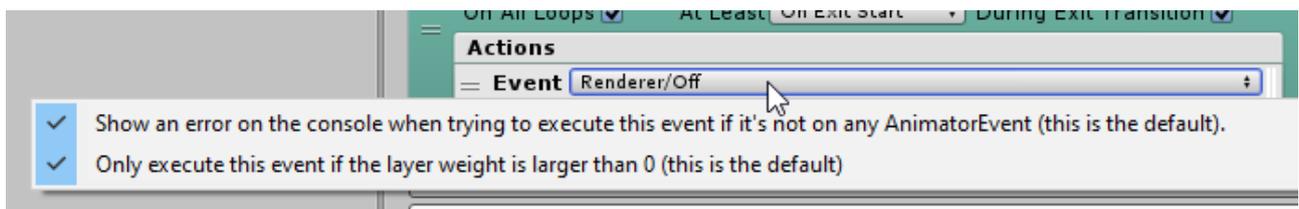


SendMessage

Check out the SendMessage documentation in the Unity documentation.

Extra

Right click on an action of an **Event SMB** for extra settings.



How it works

- AnimatorEvent holds the list of all UnityEvents that can be executed by EventSMB.
- Each time the Animator is updated, for each of its layers (from top to bottom):
 - For each state (from oldest playing to newest, there are multiple playing when playing a transition) (exit events will always be performed before enter events):
 - For each **Event SMB** in the state (from top to bottom):
 - For each entry in the **Event SMB** (from top to bottom):
 - Checks its conditions. If both the default conditions and the extra conditions pass:
 - Execute its actions immediately (this means it can affect the conditions of next entries).

TIPS

- You can use "On Normalized Time Reached" together with "At Least: On Exit Start" to make sure an event is executed.
- You can change the events while playing to experiment, but they won't be saved.
- You can add several "AnimatorEvent" components, this can be useful when using nested prefabs.
- Using a normalized time greater than one means waiting until the state loops the amount of times specified.
- Changing the name of an animation parameter may break events that rely on it. Set the names properly before using them with EventSMB.
- Use the letter semicolon ";" on the name of saved condition groups to group them in a folder structure on the "Load" dropdown.

This asset can be placed in any folder (as long as it's not called "Editor").