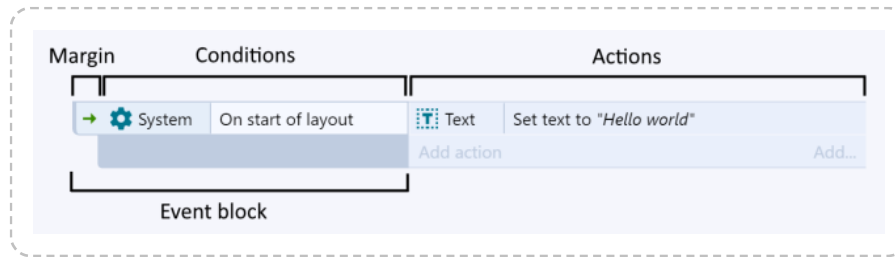


CONDITIONS

View online: <https://www.construct.net/en/make-games/manuals/construct-3/project-primitives/events/conditions>

In events, **conditions** filter instances that meet some criteria. They appear to the left of the event. All conditions in an event must be met by at least one instance for the **actions** to run. The actions then only apply to the instances that met the conditions.



System conditions do not pick any instances: they are simply either true or false, unless they specifically reference an object, such as with *Pick random instance*.

There are three kinds of conditions: normal conditions, **triggered** conditions, and **looping** conditions. You can also create OR blocks which run if *any* condition is true, rather than all the conditions. See [How events work](#) for more information.

Adding and editing conditions

When you add a new event, you are taken through the process of adding the first condition for the event. This is described in more detail in the [Event Sheet View](#) manual entry.

You can add multiple conditions to an event block. To add another condition, right-click either an existing condition or the event margin and select Add another condition. All conditions must be met for the event to run, unless you set the event to be an OR block, in which case *any* condition can be true for the event to run. To set an OR block, right-click the event margin and select Make OR block.

To edit a condition, **double-click** it. You can also **right-click** it and choose Replace or Delete.

Inverting conditions

Conditions can be **inverted**, which flips the thing they test. For example, the condition *Monster is overlapping Player* is true whenever a monster is touching the player. However, if inverted, it appears with a red invert icon and means *Monster is **not** overlapping Player*.



Not all conditions can be inverted (e.g. triggers can't be, because the event doesn't make sense inverted in that case).

Breakpoints

Paid plans only It is possible to place a breakpoint on a condition, to pause execution when it is reached in the [debugger](#). For more information, see [breakpoints](#).