

BREAKPOINTS

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

Paid plans only Breakpoints are an advanced feature that allow you to pause execution of the event sheet on a specific **event**, **condition** or **action** when running in the **debugger**. This can be a significant aid to debugging, since the full capabilities of the debugger can be used while stepping through events, conditions and actions one-by-one.

Setting a breakpoint

Breakpoints can be toggled on and off for the selected event block, condition or action by **right-clicking** them and selecting Toggle breakpoint or pressing the **F3** keyboard shortcut. Breakpoints can also be toggled while debugging.

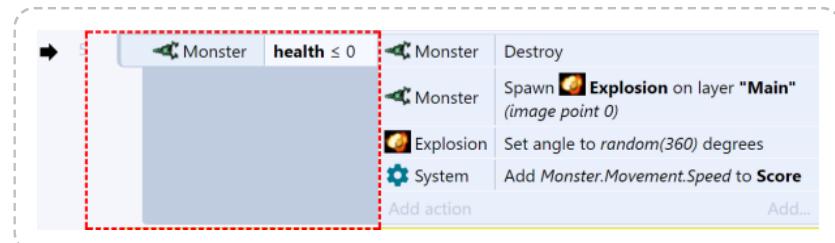
When a breakpoint is set on an event, condition or action, a breakpoint icon appears beside it.



Pausing on a breakpoint

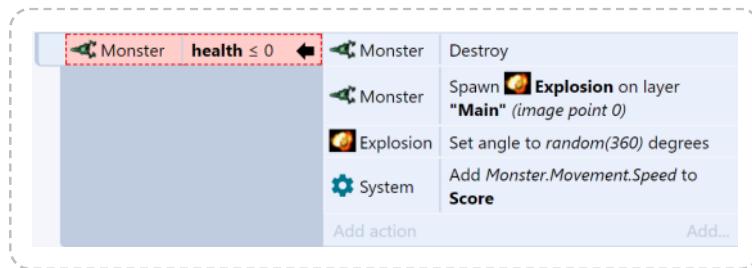
When running the debugger, the project will automatically pause *just before* it runs an event, condition or action with a breakpoint.

For events, this means it pauses just before it tests the first condition. This means a top-level event with a breakpoint will pause every tick, since the event engine reaches it every tick to test its conditions. It is usually more useful to place event breakpoints on **sub-events**, since they will only pause when the parent events have been run. When paused on a breakpoint, the event has a dashed outline and the icon changes to an arrow.

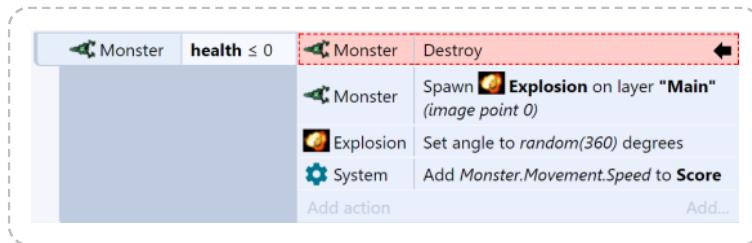


For conditions, this means it pauses just before it tests the condition. Since the condition has not yet been tested, the debugger does not know whether the condition will return true or false; you must resume execution to be able to tell. Note that Construct bails out of events early if any

condition is false. This means a breakpoint on the second condition won't pause the project if the first condition is false, since Construct will immediately skip to checking the next event. When paused on a condition, it is also indicated with a dashed outline, an arrow and also a changed background color.



For actions, this means it pauses just before the action runs. Placing a breakpoint on the first action in an event is often more useful than placing a breakpoint on the event itself, since it will only pause when all the conditions have been met and before any actions have run, as opposed to just before it starts checking any conditions. When paused on an action, it is indicated similarly to a condition.



Continuing execution

When paused on a breakpoint, the debugger can be used as normal to inspect or edit values. However the **Pause** and **Step** buttons change in to **Continue** and **Next**.

Pressing **Continue** resumes until the next breakpoint is hit. Remember you can still toggle other breakpoints while paused on a breakpoint.

Pressing **Next** will step to the next event, condition or action in sequence in the event sheet. Alternatively, press **F10** with the browser window focused. This is useful for stepping through an event action-by-action and watching what happens in the debugger's inspector.

Limitations

Unfortunately due to the architecture of the Construct engine, breakpoints cannot be placed inside some (but not all) triggered events. When not supported, this also means breakpoints cannot be anywhere inside sub-events to the triggered event.

Using breakpoints

Breakpoints can be incredibly useful to learn more about how your own events work and to help track down problems. They may take some getting used to in order to get the most out of them, but once familiar they can be indispensable.

It is especially useful to have a multi-monitor setup when using breakpoints, or with the debugger in general. This allows you to have Construct showing which event it has reached on one screen, and the project running on another screen.