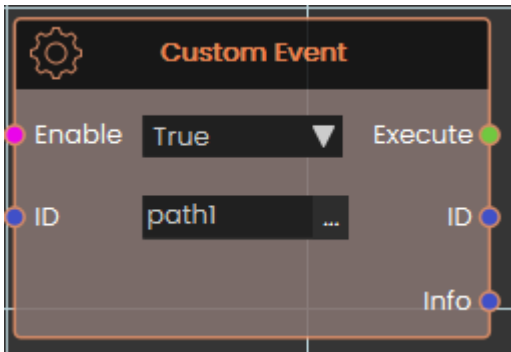


Execution

Custom Event



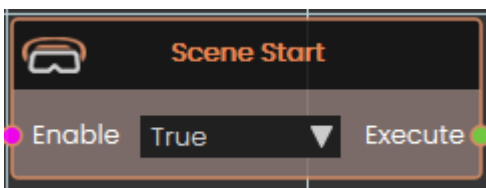
This event is activated manually by the user. It's useful when you want multiple different paths or actions to lead to the same response.

To set it up, use the **Trigger Custom Event** response and assign it a unique ID (e.g., `path1`). You can place this trigger in as many locations as needed.

Whenever a **Trigger Custom Event** is called, it will execute the **Custom Event** that has a matching ID.

If you want to pass additional info, you can use the info field in **Trigger Custom Event** response, this data will be passed to the even in the info port

Scene Start

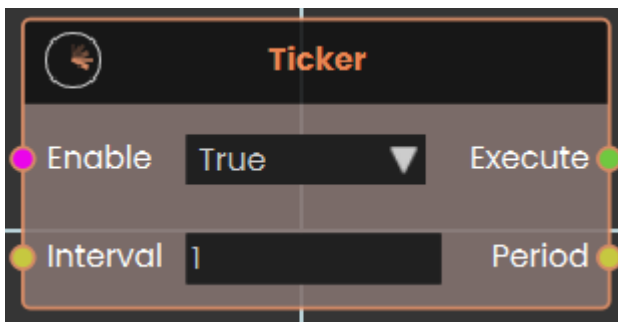


This event is triggered automatically when the scene begins. Use it to initialize elements at startup, such as playing videos, running background animations, or any other setup

tasks.

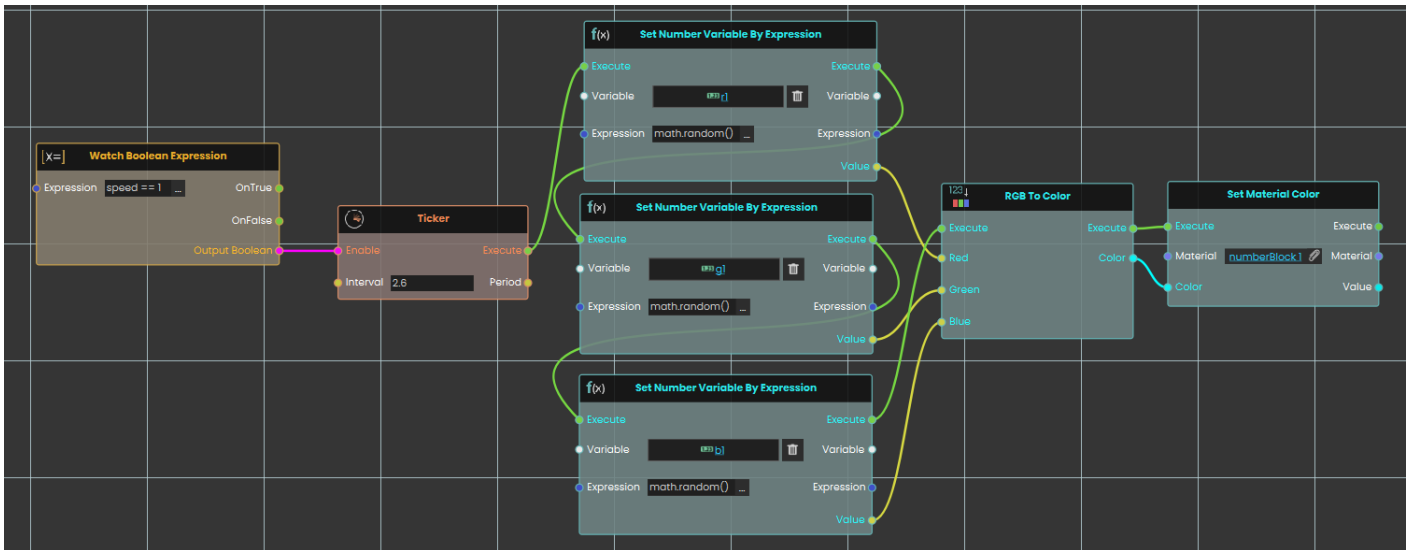
Note: Since this event fires the moment the scene loads, it's recommended to add a small delay before making additional scene changes — this ensures all scene elements have fully initialized before anything else runs.

Ticker



The **Ticker** event node functions as a continuous, time-based loop that repeatedly activates subsequent nodes at a consistent rate. As long as its Enable input is set to *True*, the node constantly triggers its Execute output every time the duration specified in the numerical Interval input (measured in seconds) elapses. Once activated, it continually outputs this time value through the Period pin, allowing the logic flow to perform recurring actions or background checks for as long as the node remains enabled.

Example:



In this example, we use the Ticker node to create a loop that randomly changes a material's color every 2.5 seconds:

1. A **Watch Boolean Expression** node continuously monitors the "speed1" variable and activates the **Ticker** node once the value equals 1.
2. While enabled, the **Ticker** node creates a continuous loop that triggers its execution output every 2.5 seconds.
3. The Ticker simultaneously executes three **Set Number Variable By Expression** nodes to generate random values using the `math.random()` function for the variables r1, g1, and b1.
4. These three random numbers are passed into an **RGB To Color** node, where they are combined to formulate a new color.
5. Finally, this new color is sent to a **Set Material Color** node, which immediately applies the randomly generated color to the target material.

Revision #7

Created 17 March 2026 18:05:11 by Ashraf Sultan

Updated 31 March 2026 14:56:47 by Ahmad Qasim