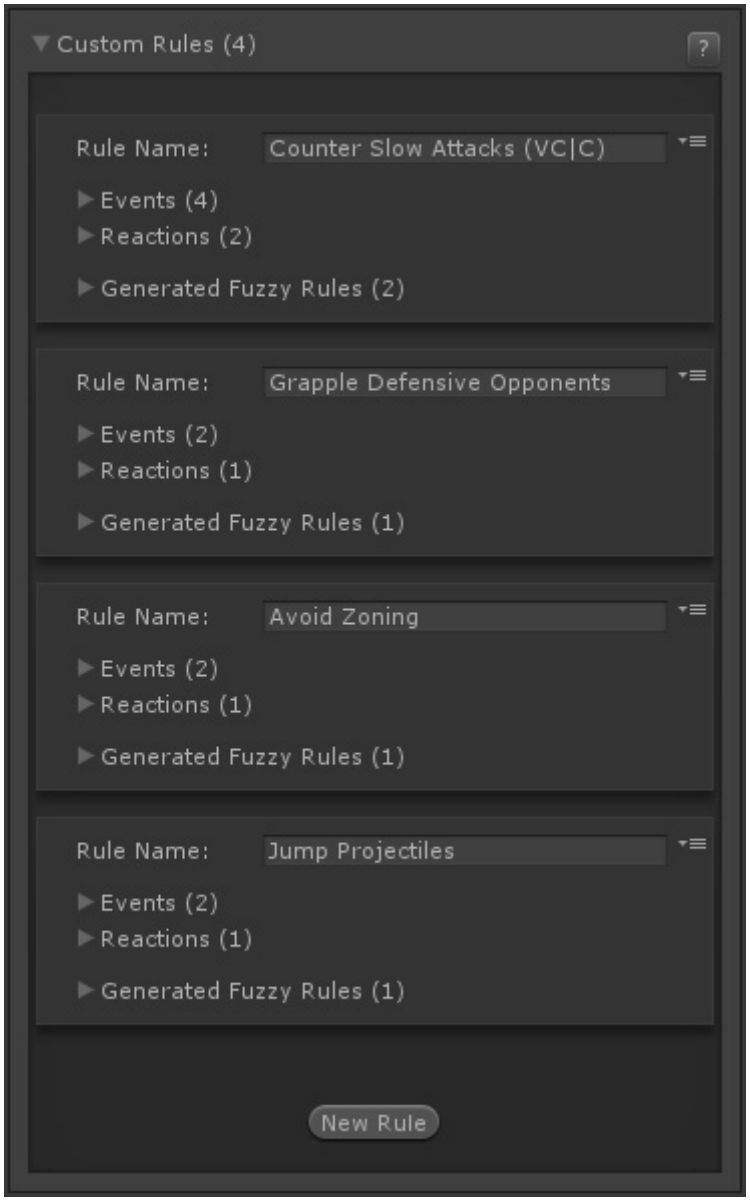
Custom Rules

Here you can design your own fuzzy rules with an intuitive logic process. Fuzzy Logic operates with Linguisit Variables to facilitate the weight calculation process.

This process uses `.\UFE Addons\Runtime\AIRulesGenerator.cs` to automatically generates its own fuzzy rules.



Rule

A rule is a set of events and conditions that tell the engine how likely a reaction should be.

This interface was created to better emulate linguist variables into UFE through its events and conditions. From a logical operator point of view, think of events, conditions and reactions with the following operators:

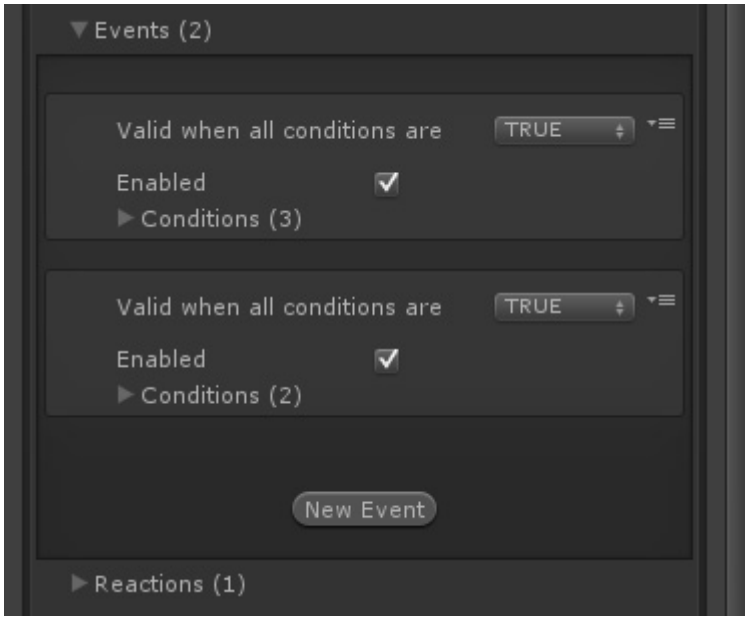
```
IF (Condition1 AND Condition2 AND …) OR (Condition3 AND Condition4 AND …)
THEN [Reaction] is [Desirable Value]
```

Where:
`(Condition1 AND Condition2 AND …)` Equals `Event1`
`(Condition3 AND Condition4 AND …)` Equals `Event2`
And so on…

The linguistic variables used by the engine are a direct translation of this system, and they can be seen by clicking on *Generated Fuzzy Rules*.

By default, every reaction has 0 weight, and by default, if no weight is applied to reactions, the character should stay idle.

Rule Name: Use this to make a small description of what the rule is suppose to do.

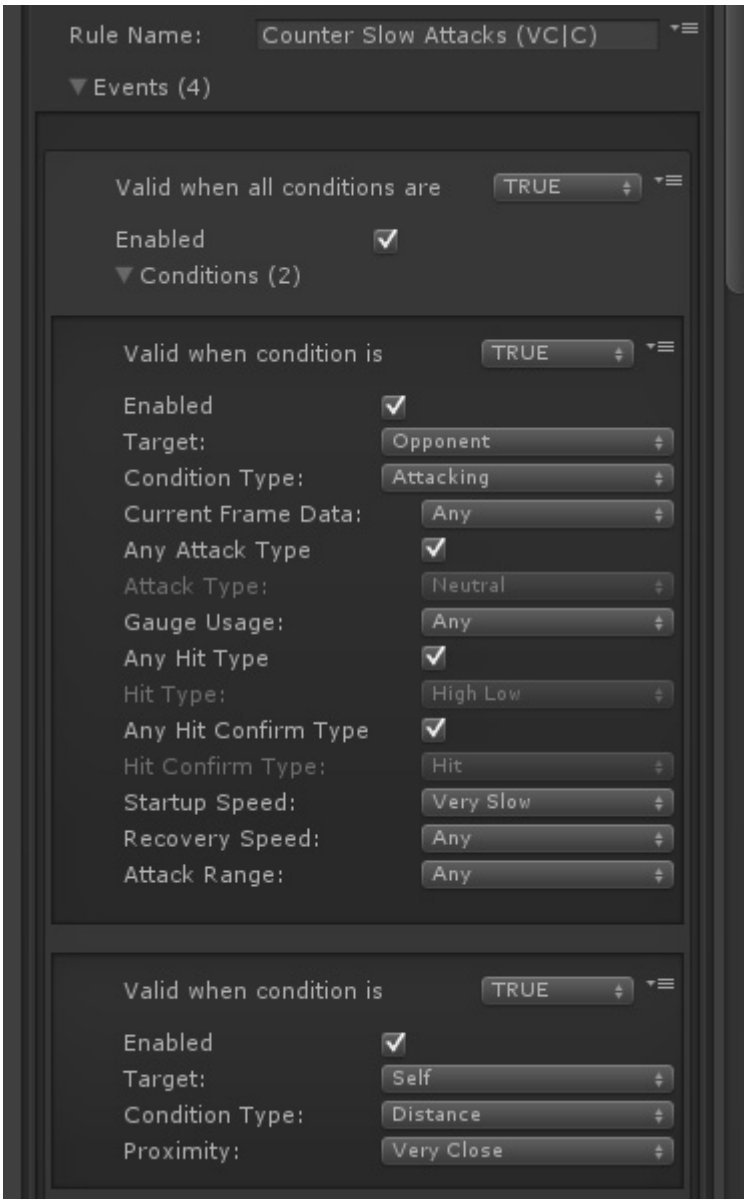


Event

Events contain conditions. A Rule is valid when at least one of the events is **valid**.

Valid when all conditions are: Allow you to invert the logic of this event. If set to *FALSE*, this event will be considered as a valid entry for the Reaction weight manipulation if its not *TRUE*.

Enabled If disabled, this event will not be considered for validation. Use this option for debugging purposes.



Event: Condition

An event is only valid if all of its conditions are **valid**.

Valid when condition is: Allow you to invert the logic of this condition. If set to *FALSE*, this condition will be considered to be valid if the dictated entry is not *TRUE*.

Enabled If disabled, this condition will not be considered for validation. Use this option for debugging purposes.

Target: Should this condition happen to the opponent or to itself.

Condition Type:

- *Idle*: Condition valid if the target is idle.
- *Horizontal Movement*: Condition valid if the target is moving horizontally.
  - Direction

        - *Moving Forward*: is moving forward.
        - *Sill*: is not moving horizontally.
        - *Moving Back*: is moving back.
    - Movement Speed: The speed the target is moving (numeric variations can be defined under Definitions).

- *Vertical Movement*: Condition valid if the target is moving vertically.
    - Direction
        - *Crouching*: is crouching.
        - *Standing*: is not moving vertically.
        - *Jumping*: is jumping (or moving vertically through applied force).

- *Health Status*: Condition valid if the target's health matches the `Health` variation set here.

- *Gauge Status*: Condition valid if the target's gauge/meter matches the `Gauge` variation set here.

- *Distance*: Condition valid if the target's distance matches the `Proximity` variation set here (numeric variations can be defined under Definitions).

- *Attacking*: Condition valid if target is playing a move matching the conditions below.
    - Current Frame Data: Valid for moves with active frames.
        - *Startup Frames*: move being played is on its startup frames.
        - *Active Frames*: move being played is on its active frames.
        - *Recovery Frames*: move being played is on its recovery frames.
    - Attack Type: Move matches the classification provided under Move Editor -> AI Definitions.
    - Gauge Usage: Move uses a certain amount of gauge/meter.
    - Hit Type: Move matches this classification under Move Editor -> AI Definitions.
    - Hit Confirm Type: Move matches this classification under Move Editor -> AI Definitions.
    - Startup Speed: Move matches this classification under Move Editor -> AI Definitions.
    - Recovery Speed: Move matches this classification under Move Editor -> AI Definitions.
    - Attack Range: Move matches this classification under Move Editor -> AI Definitions.

- *Blocking*: Condition valid if target is blocking.
    - Blocking State: Is the target blocking High|Low|Air.

- *Stunned*: Condition valid if target is stunned.

- *Down*: Condition valid if target is down.

Reaction

When the conditions for one of the events is true, how likely is the AI to attempt the following reaction.

Generated Fuzzy Rules

Allows you to read the result of all the rules being generated by these options. Useful to debug how each combination interact and the end result that is sent to the Fuzzy Core.

This option is for debug viewing only and it has no impact on the game.

< Back to A.I. Editor