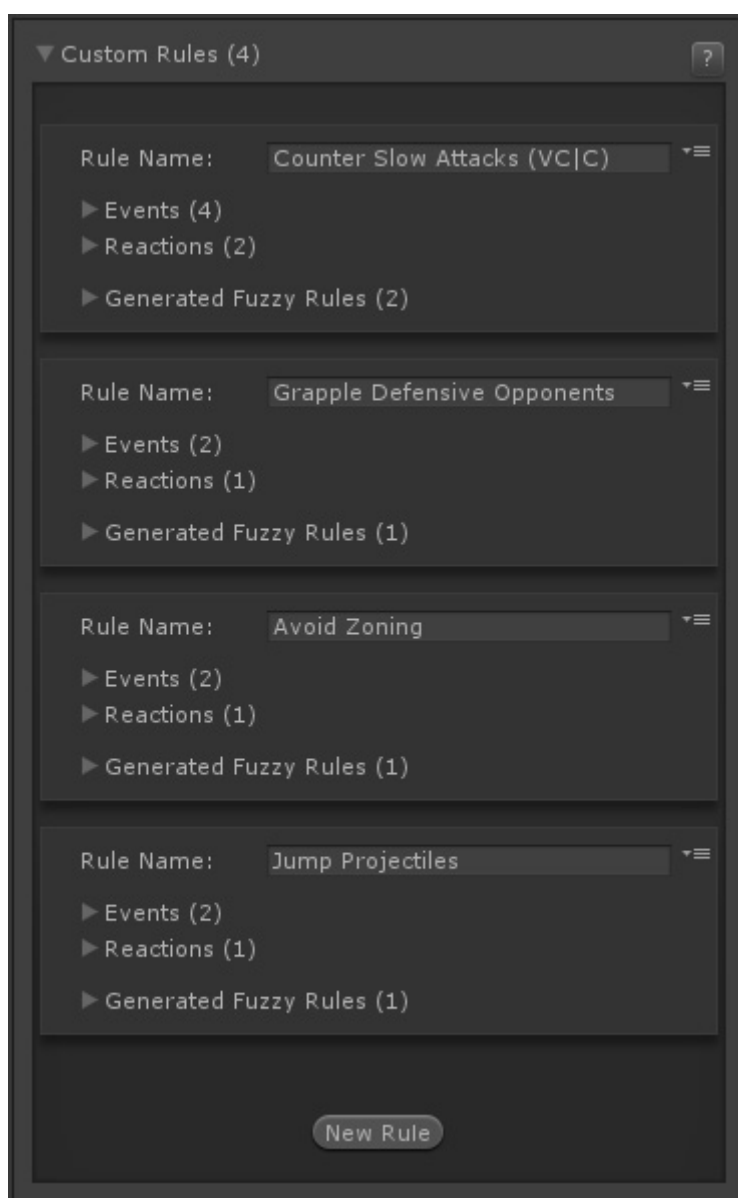


Custom Rules

Here you can design your own fuzzy rules with an intuitive logic process. Fuzzy Logic operates with [Linguistic 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 [Desirability Value]

Where:

(Condition1 AND Condition2 AND ...) Equals Event1

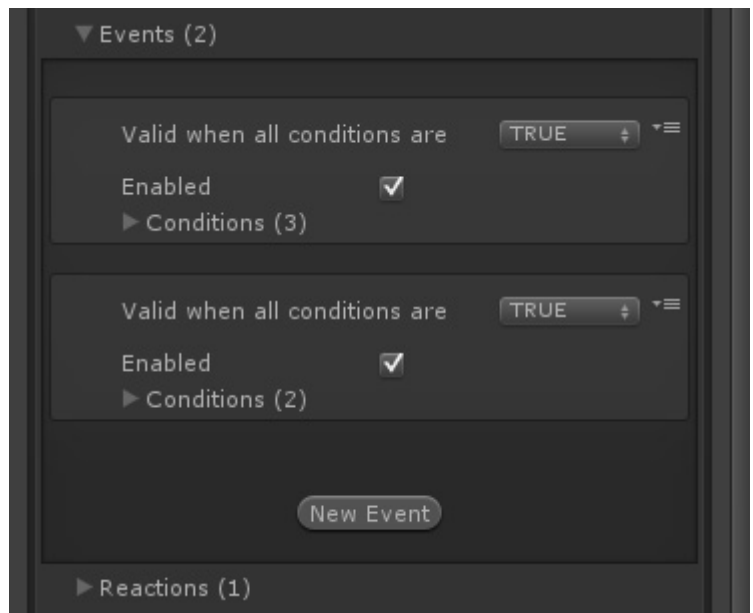
(Condition3 AND Condition4 AND ...) Equals Event2

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 active 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.

Rule Name: Counter Slow Attacks (VC|C)

▼ Events (4)

Valid when all conditions are TRUE

Enabled ☒

▼ Conditions (2)

Valid when condition is TRUE

Enabled ☒

Target: Opponent

Condition Type: Attacking

Current Frame Data: Any

Any Attack Type ☒

Attack Type: Neutral

Gauge Usage: Any

Any Hit Type ☒

Hit Type: High Low

Any Hit Confirm Type ☒

Hit Confirm Type: Hit

Startup Speed: Very Slow

Recovery Speed: Any

Attack Range: Any

Valid when condition is TRUE

Enabled ☒

Target: Self

Condition Type: Distance

Proximity: Very Close

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.
 - *Still*: is not moving horizontally.
 - *Moving Back*: is moving back.
- Movement Speed: The speed the target is moving (numeric values for this option 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 (numeric values for this option can be defined under [Definitions](#)).
- *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 values for this option 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 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.

The screenshot shows a configuration panel titled "Reactions (3)". It contains three reaction entries, each with a set of conditions and a reaction type.

- Reaction 1:**
 - Reaction Type: Crouch
 - Desirability: Desirable
- Reaction 2:**
 - Reaction Type: Play Move
 - Any Attack Type: ☒
 - Attack Type: Neutral
 - Gauge Usage: Any
 - Any Hit Type: ☒
 - Hit Type: High Low
 - Any Hit Confirm Type: ☒
 - Hit Confirm Type: Hit
 - Startup Speed: Fast
 - Recovery Speed: Any
 - Attack Range: Any
 - Desirability: Very Desirable
- Reaction 3:**
 - Reaction Type: Play Move
 - Any Attack Type: ☒
 - Attack Type: Neutral
 - Gauge Usage: Any
 - Any Hit Type: ☒
 - Hit Type: High Low
 - Any Hit Confirm Type: ☒
 - Hit Confirm Type: Hit
 - Startup Speed: Very Fast
 - Recovery Speed: Any
 - Attack Range: Any
 - Desirability: The Best Option

At the bottom of the panel is a button labeled "New Reaction".

Reaction Type: Tells the AI to execute the following command

- *Idle*: Stop pressing inputs.
- *Move Forward/Back*: Attempts to move forward/back.
- *Crouch*: Attempts to crouch.
- *Jump Straight/Forward/Back*: Attempts to jump Straight/Forward/Back.
- *Crouch Block*: Attempts to crouch and block at the same time.
- *Stand Block*: Attempts to block while standing. If your game is set to have "hold back" as potential block, the character will also move backwards.

- *Jump Block*: Attempts to block while jumping.
- *Play Move*: Attempts to play a move with the following characteristics:
 - Attack Type: Move matches the classification 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](#).
- *Change Behaviour*: The AI will attempt to load a new instruction file based on the behaviour selected. Only works if the character has an instruction set for that behaviour.

Desirability: Tells the AI how desirable (or not) this reaction is (numeric values for this option can be defined under [Definitions](#)).

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](#)

From:
<http://www.ufe3d.com/> - **Universal Fighting Engine**

Permanent link:
<http://www.ufe3d.com/doku.php/ai:customrules?rev=1414429843>

Last update: **2014/10/27 13:10**

