RuleBasedGodot: a Rule-Based System for Godot game engine

Rodrigo Volpe Battistin | June 17, 2023

RuleFormatSaver
- save(resource: Resource, path: String, flags: int) -> Error

RuleFormatLoader
- load(path: String, original_path: String, use_sub_threads: bool, cache_mode: int)

RuleBasedSystem
+ rule_set: RuleSet
+ rule_arbitration: Enum
- arbiter: Arbiter

+ iterate() -> Array

RuleSet
- rules: Array[Rule]

+ satisfied_rules() -> Array[Rule]

**<>
RuleBasedResource**
- system_node: RuleBasedSystem

+ setup(system_node: RuleBasedSystem) -> void
+ to_json_string() -> String
+ build_from_repr(json_repr) -> void
- eval_arguments(type_to_value: Dictionary) -> Dictionary
- eval_value(type: StringName, value: String) -> Variant

**<>
Arbiter**
select_rule_to_trigger(satisfied_rules: Array[Rule]) -> Rule

FirstApplicableArbiter

LeastRecentlyUsedArbiter

Rule
+ condition: Match
+ actions: Array[Action]

+ trigger_actions() -> Array

RuleFactory
+ create_rule(json_repr: Dictionary) -> Rule
+ create_action(json_repr: Array) -> Action
+ create_match(json_repr: Array) -> Match

**<>
AbstractAction**
+ trigger() -> Variant

SetPropertyAction

CallMethodAction

EmitSignalAction

**<>
AbstractMatch**
+ is_satisfied() -> bool

NOTMatch

ANDMatch

ORMatch

StringMatch

NumericMatch

DistanceMatch

AreaDetectionMatch