HuNavSim 2.0 – Behaviour-Tree Nodes Reference

Table of Contents

Units

Distances in **metres [m]** · Angles in **radians [rad]** · Times in **seconds [s]**

---

1 Simple Condition Nodes

Node	Purpose
RandomChanceCondition	Succeeds probabilistically based on a given chance
IsRobotFacingAgent	Checks if the robot is oriented towards the agent
IsAgentVisible	Determines if a target agent is visible to an observer
IsRobotClose	Evaluates whether the robot is within a threshold distance of the agent
IsAgentClose	Determines whether one agent is within a close distance of another agent
IsAtPosition	Checks if the agent has reached a specified goal position within a tolerance

1.1 RandomChanceCondition

Description
Succeeds probabilistically based on a given chance.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `probability` (*double*): Chance of success.

1.2 IsRobotFacingAgent

Description
Checks if the robot is oriented towards the agent.

Inputs

- `agent_id` (*int*): Identifier of the agent.

1.3 IsAgentVisible

Description
Determines if a target agent is visible to an observer.

Inputs

- `observer_id` (*int*): Identifier of the observing agent.
- `agent_id` (*int*): Identifier of the target agent.
- `distance` (*double*): Visibility distance threshold.

1.4 IsRobotClose

Description
Evaluates if the robot is within close proximity to the agent, defined by a threshold.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `threshold` (*double*): Distance threshold for close proximity.

1.5 IsAgentClose

Description
Determines whether one agent is within a close distance to another agent.

Inputs

- `observer_id` (*int*): Identifier of the observing agent.

- `target_agent_id` (*int*): Identifier of the target agent.

1.6 IsAtPosition

Description
Checks if the agent has reached a specified target goal position within a given tolerance.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `goal_id` (*int*): Target goal ID.
- `tolerance` (*double*): Acceptable tolerance for reaching the target.

---

2 Simple Action Nodes

Node	Purpose
FindNearestAgent	Identifies the nearest agent relative to a given agent
SaySomething	Commands the agent to publish a ROS message
SetGroupId	Sets the group identifier for the agent
SetGoal	Establishes a navigation target by setting a goal position
StopMovement	Commands the agent to immediately halt all movement
ResumeMovement	Instructs the agent to resume movement after being stopped

2.1 FindNearestAgent

Description
Identifies the nearest agent relative to a given agent.

Inputs

- `agent_id` (*int*): Identifier of the agent searching for the nearest target.

Outputs

- `target_agent_id` (*int*): Identifier of the nearest agent found.

2.2 SaySomething

Description
Commands the agent to publish a ROS message.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `message` (*string*): The message to be published.

2.3 SetGroupId

Description
Sets the group identifier for the agent.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `group_id` (*int*): New group identifier.

2.4 SetGoal

Description
Establishes a navigation target by setting a goal position.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `goal_id` (*int*): Goal ID of the target goal.

2.5 StopMovement

Description
Commands the agent to immediately halt all movement (stay idle).

Inputs

- `agent_id` (*int*): Identifier of the agent.

2.6 ResumeMovement

Description
Instructs the agent to resume its movement after being stopped.

Inputs

- `agent_id` (*int*): Identifier of the agent.

---

3 Stateful Action Nodes

Node	Purpose
StopAndWaitTimerAction	Stop-and-wait behaviour for a defined duration
ConversationFormation	Manages the formation of a conversation among multiple agents
GoTo	Commands the agent to navigate directly to a specified point
ApproachAgent	Directs the agent to move towards another agent for a defined duration
ApproachRobot	Commands the agent to approach the robot for a defined duration
BlockRobot	Instructs the agent to block the robot's path for a specified duration
BlockAgent	Commands the agent to block another agent's path for a defined duration
GroupWalk	Directs a group to walk together with a designated main agent
LookAtPoint	Makes the agent orient towards a specific point in space
LookAtAgent	Directs an observer agent to focus on another agent
LookAtRobot	Commands the agent to direct its attention toward the robot
FollowAgent	Commands an agent to follow another target agent

3.1 StopAndWaitTimerAction

Description
Implements a stop-and-wait behaviour that stops the agent for a defined duration.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `wait_duration` (*double*): Duration for which the agent should wait.

3.2 ConversationFormation

Description
Manages the formation of a conversation among multiple agents.

Inputs

- `main_agent_id` (*int*): Identifier of the primary agent leading the conversation.
- `conversation_duration` (*double*): Total duration of the conversation.
- `goal_id` (*int*): Goal ID where the conversation's central point will take place.
- `time_step` (*double*): Time step for movement updates.
- `non_main_agent_ids` (*string*): Comma-separated list of participating agent IDs.

3.3 GoTo

Description
Commands the agent to navigate directly to a specified point.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `time_step` (*double*): Time step for movement updates.
- `goal_id` (*int*): Chosen goal ID to go to.
- `tolerance` (*double*): Distance [m] to consider "at goal".

3.4 ApproachAgent

Description
Directs the agent to move towards another agent for a defined duration.

Inputs

- `agent_id` (*int*): Identifier of the approaching agent.
- `target_agent_id` (*int*): Identifier of the target agent.
- `time_step` (*double*): Time step for movement updates.
- `closest_dist` (*double*): Distance at which the agent is considered to have approached sufficiently.
- `max_vel` (*double*): Maximum velocity for the approach.
- `duration` (*double*): Duration of the approach action.

3.5 ApproachRobot

Description
Commands the agent to approach the robot for a defined duration.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `time_step` (*double*): Time step for movement updates.
- `closest_dist` (*double*): Distance considered close enough to the robot.
- `max_vel` (*double*): Maximum velocity during the approach.
- `duration` (*double*): Duration of the approach behaviour.

3.6 BlockRobot

Description
Instructs the agent to block the robot's path for a specified duration.

Inputs

- `agent_id` (*int*): Identifier of the agent.

- `time_step` (*double*): Time step for movement updates.
- `front_dist` (*double*): Distance in front of the agent used for blocking.
- `duration` (*double*): Duration for which the agent will block.

3.7 BlockAgent

Description
Commands the agent to block another agent's path for a defined duration.

Inputs

- `agent_id` (*int*): Identifier of the blocking agent.
- `target_agent_id` (*int*): Identifier of the agent to be blocked.
- `time_step` (*double*): Time step for movement updates.
- `front_dist` (*double*): Blocking distance threshold.
- `duration` (*double*): Duration of the blocking action.

3.8 GroupWalk

Description
Directs a group of agents to walk together with a designated main agent leading and the others following along.

Inputs

- `main_agent_id` (*int*): Identifier of the main agent guiding the group.
- `time_step` (*double*): Time increment used for updating movement.
- `non_main_agent_ids` (*string*): Comma-separated list of the non-main agents' identifiers.
- `duration` (*double*, optional): Duration for which the behaviour runs. If omitted, the behaviour runs indefinitely.

3.9 LookAtPoint

Description
Makes the agent orient towards a specific point in space.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `goal_id` (*int*): Goal ID of the point to look at.
- `yaw_tolerance` (*double*): Angle tolerance [rad] to consider "aligned".

3.10 LookAtAgent

Description
Directs an agent (acting as the observer) to focus on another agent.

Inputs

- `observer_id` (*int*): Identifier of the observing agent.
- `target_id` (*int*): Identifier of the target agent.
- `yaw_tolerance` (*double*): Angle tolerance [rad] to consider "aligned".

3.11 LookAtRobot

Description
Commands the agent to direct its attention toward the robot.

Inputs

- `agent_id` (*int*): Identifier of the agent.
- `yaw_tolerance` (*double*): Angle tolerance [rad] to consider “aligned”.

3.12 FollowAgent

Description
Commands an agent to follow another target agent.

Inputs

- `agent_id` (*int*): Identifier of the follower agent.
- `time_step` (*double*): Time step for movement updates.
- `target_agent_id` (*int*): Identifier of the agent to be followed.
- `duration` (*double*, optional): Duration for which the behaviour is active. If omitted, the behaviour runs indefinitely.

---

4 Stateful Condition Nodes

Node	Purpose
IsAnyoneSpeaking	Evaluates whether any agent within a given distance is speaking
IsSpeaking	Checks whether a specified agent is speaking
IsAnyoneLookingAtMe	Determines if any agent is looking at the agent
IsLookingAtMe	Checks whether a specified target agent is looking at the agent

4.1 IsAnyoneSpeaking

Description
For a specified duration, evaluates whether any agent within a given distance is speaking (publishing a ROS-2 string message).

Inputs

- `agent_id` (*int*): Identifier of the agent checking for speakers.
- `time_step` (*double*): Time step for movement updates.
- `distance_threshold` (*double*): Maximum distance within which speaking is detected.
- `duration` (*double*): Duration over which the speaking condition is evaluated.

Outputs

- `speaker_id` (*int*): Identifier of the detected speaking agent.

4.2 IsSpeaking

Description
Checks whether the specified agent is actively speaking within a set distance threshold over a specified duration.

Inputs

- `agent_id` (*int*): Identifier of the agent checking for the speaker.
- `time_step` (*double*): Time step for movement updates.
- `target_id` (*int*): Identifier of the agent whose speaking status is evaluated.
- `distance_threshold` (*double*): Distance range for detecting the speaking condition.
- `duration` (*double*): Duration over which the condition is considered.

4.3 IsAnyoneLookingAtMe

Description
Determines if any agent is looking at the agent and returns the observer's identifier if the condition is met.

Inputs

- `agent_id` (*int*): Identifier of the agent being observed.
- `time_step` (*double*): Time step for movement updates.
- `distance_threshold` (*double*): Maximum distance for detecting potential observers.
- `angle_threshold` (*double*): Angular threshold specifying how precisely the observer must be aligned.
- `duration` (*double*): Duration that the condition must persist.

Outputs

- `observer_id` (*int*): Identifier of the agent that is looking at the target agent.

4.4 IsLookingAtMe

Description
Checks whether a specified target agent is looking at the agent over a defined time, distance, and angle.

Inputs

- `agent_id` (*int*): Identifier of the agent being looked at.
- `time_step` (*double*): Time step for movement updates.
- `target_id` (*int*): Identifier of the target agent that is potentially looking.
- `distance_threshold` (*double*): Maximum range to detect the look direction.
- `angle_threshold` (*double*): Angular threshold for the looking condition.
- `duration` (*double*): Duration over which the condition is checked.

---

5 Decorator Nodes

Node	Purpose
TimeDelayDecorator	Delays the execution of its child node by a specified amount of time

5.1 TimeDelayDecorator

Description
Delays the execution of its child node by a specified amount of time. Until the delay has elapsed, this decorator returns **FAILURE**. Once the delay period is over, it ticks its child and returns the child's status.

Inputs

- `delay` (*double*, default = 1.0): Delay time in seconds before the child node is ticked.
