Angle_Check_A

Node Type

Action Node

Description

Checks if the angle between the transform and a specified target (Vector3, Transform, or GameObject) meets a certain comparison condition. It can ignore height for 2D plane checks.

Returned State

This node returns Fail if the specified field in the Blackboard is null, otherwise, it returns the comparison condition result.

Required Components

None

Back to top

Angle_Check_B

Node Type

Branching Node

Description

Checks if the angle between the transform and a specified target (Vector3, Transform, or GameObject) meets a certain comparison condition. It can ignore height for 2D plane checks. If the condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children.

Returned State

This node returns Fail if the specified field in the Blackboard is null or the comparison condition is not met, otherwise, it returns the update result of the child nodes.

Required Components

None

Back to top

Animator_Check

Node Type

Branching Node

Description

Evaluates whether the current animation state of the specified animator matches the required state name. If the condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children. This node returns Failure if the Animator is null or if the current animation state does not match the required name, otherwise, it returns the update result of the child nodes.

Returned State

This node returns Fail if the specified field in the Blackboard is null or the comparison condition is not met, otherwise, it returns the update result of the child nodes.

Required Components

Animator component attached to executing GameObject.

Back to top

Animator_Set

Node Type

Action Node

Description

Sets multiple parameters (float, int, bool, and trigger) on the Animator based on predefined keys and values. Ensures that all required parameters are declared in the Animator before updating.

Returned State

This node returns Fail if the Animator is null or if any of the specified parameters are not declared in the Animator.

Required Components

Animator component attached to executing GameObject.

Back to top

BoxCollider_Set

Node Type

Action Node

Description

Updates the size and center of a specified BoxCollider based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs). Allows for update based on various entry types, including constants, random ranges, and animation curves.

Returned State

This node returns Failure if the BoxCollider is null.

Required Components

BoxCollider.

Back to top

CapsuleCollider_Set

Node Type

Action Node

Description

Updates the size, radius and height of a specified CapsuleCollider based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs). Allows for update based on various entry types, including constants, random ranges, and animation curves.

Returned State

This node returns Failure if the CapsuleCollider is null.

Required Components

CapsuleCollider.

Back to top

Collider_isTrigger

Node Type

Action Node

Description

Changes the isTrigger state of a specified Collider (enabled, disabled, or toggled from its current state) based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs).

Returned State

This node returns Failure if the Collider is null.

Required Components

Collider.

Back to top

Collider_OnCollision

Node Type

Action Node

Description

Allows monitoring the collision state of the specified Collider based on the declared type (use Blackboard for Scene Object or Node for Prefabs). Allows for choosing an OnCollisionType (OnCollisionEnter, OnCollisionExit, OnCollisionStay). Moreover, it allows filtering collisions based on the TAG attached to the collided object or a component attached to it.

Returned State

This node returns Failure if the Collider is null. Returns Failure when "Look For MonoBehaviour" is not chosen, when "Look For TAG" is not declared, or when the selected OnCollisionType does not occur.

Required Components

Collider.

Back to top

Collider_SetState

Node Type

Action Node

Description

Changes the active state of a specified Collider (enabled, disabled, or toggled from its current state) based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs).

Returned State

This node returns Failure if the Collider is null.

Required Components

Collider.

Back to top

Animator_Sync

Node Type

Action Node

Description

Synchronizes Animator parameters (float and int) based on transform's directional values like Forward, Back, Left, Right, Up, Down in specific axes.

Returned State

This node returns Fail if the Animator is null or if any specified parameters are not declared in the Animator.

Required Components

Animator component attached to executing GameObject.

Back to top

Cooldown

Node Type

Decorator Node

Description

Limits the execution of its child node by applying a cooldown period between repetitions, only allowing execution once the cooldown time has passed.

Returned State

This node returns Fail if the cooldown period has not yet expired.

Required Components

None

Back to top

Cooldown_Timer

Node Type

Decorator Node

Description

Alternating timers and cooldowns for its child node, executing tasks based on specified durations.

Returned State

Node returns Fail during cooldown durations.

Required Components

None

Back to top

DebulLog

Node Type

Action Node

Description

Logs a specified message to the console during execution.

Returned State

Node always returns Success after logging the message.

Required Components

None.

Back to top

Destroy

Node Type

Action Node

Description

Destroys the object referenced by Blackboard.This if it exists, performing a null check before destruction.

Returned State

This node returns Failure if Blackboard.This is null; otherwise, it always returns Success after destroying the object.

Required Components

None.

Back to top

Distance_Check_A

Node Type

Action Node

Description

Evaluates whether the distance between the object's transform and a specified target (Vector3, Transform, or GameObject) satisfies a defined comparison condition, with the option to ignore height for 2D plane checks.

Returned State

This node returns Fail if the specified field in the Blackboard is null, otherwise, it returns the comparison condition result.

Required Components

None

Back to top

Distance_Check_B

Node Type

Branching Node

Description

Evaluates whether the distance between the object's transform and a specified target (Vector3, Transform, or GameObject) satisfies a defined comparison condition, with the option to ignore height for 2D plane checks. If the condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children.

Returned State

This node returns Fail if the specified field in the Blackboard is null or the comparison condition is not met, otherwise, it returns the update result of the child nodes.

Required Components

Back to top

None

EventEmitter

Node Type

Action Node

Description

Emits a specified event to the EventManager during execution.

Returned State

This node always returns Success after emitting the event.

Required Components

None.

Back to top

EventListener

Node Type

Decorator Node

Description

Listens for a specified event and triggers the child node when the event occurs.

Returned State

This node returns Running until the event is triggered, then it executes the child node.

Required Components

None

Back to top

Fail

Node Type

Action Node

Description

Always returns a failure state when executed.

Returned State

This always returning Failure.

Required Components

None.

Back to top

Failer

Node Type

Decorator Node

Description

Updates its child node while ignoring its result, always returning a failure state.

Returned State

This node always returns Failure after executing the child node.

Required Components

None

Back to top

GameObject_SetLayer

Node Type

Action Node

Description

Changes the layer of a specified GameObject based on the given LayerMask and the declared type (use Blackboard - for Scene Object or Node - for Prefabs).

Returned State

This node returns Failure if the specified GameObject is null.

Required Components

None.

Back to top

GameObject_SetLayer

Node Type

Action Node

Description

Changes the active state of a specified GameObject (enabled, disabled, or toggled from its current state) based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs).

Returned State

This node returns Failure if the GameObject is null.

Required Components

None.

Back to top

GameObject_SetTAG

Node Type

Action Node

Description

Sets the tag of a specified GameObject based on the declared type (use Blackboard – for Scene Object or Node – for Prefabs).

Returned State

This node returns Failure if the GameObject is null.

Required Components

None.

Back to top

GameObject_Spawn

Node Type

Action Node

Description

Instantiates a specified GameObject based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs) with configurable position, rotation, and scale based on the declared entry types (Constant or Random Between Two Constants). The new GameObject TAG and layer can also be set.

Returned State

This node returns Failure if the prefab is null.

Required Components

None.

Back to top

Health_Change

Node Type

Action Node

Description

Modifies the Blackboard.Health of the character by a specified value using different override types (New Value, Change by Value, Multiply by Value).

Returned State

This node always returns Success.

Required Components

None.

Back to top

Health_Check

Node Type

Branching Node

Description

Evaluates the value of Blackboard.Health against a specified value using various comparison types (greater than, less than, etc.). If the condition is true, it executes the Left Output Children; otherwise, it executes the Right Output Children.

Returned State

This node returns the update result of the child nodes.

Required Components

None

Back to top

Health_WasChanged

Node Type

Branching Node

Description

Evaluates whether the entity's health has changed (either increased or decreased) since the last update based on the specified comparison type. If the condition is met, it executes the Left Output Children; otherwise, it executes the Right Output Children.

Returned State

This node returns the update result of the child nodes.

Required Components

None

Back to top

Inverter

Node Type

Decorator Node

Description

Inverts the result of the child node's update. This allows for more complex behavior within the node tree structure.

Returned State

If the child node returns Success, it returns Failure, and vice versa. If the child is running or slipping, it reflects that state.

Required Components

None

Back to top

Limiter

Node Type

Decorator Node

Description

Limits the number of times the child node can execute based on the specified repetition count. Each time the child node is updated, it decrements the remaining repetitions and allows the child to run until the limit is reached.

Returned State

While the limit is not reached, return the child's state. When the limit is reached, set the child's state to Failure and return Failure.

Required Components

None

Back to top

LineOfSight_Check

Node Type

Branching Node

Description

Evaluates whether a direct line of sight exists between the entity and a specified target, considering specified tags and ignoring certain colliders as defined by the LayerMask. If a clear line of sight to the target is detected, it executes the Left Output Children; otherwise, it executes the Right Output Children.

Returned State

This node returns Failure if the target is null or if the maximum iterations are reached (internal safety). Otherwise, it returns the update result of the child nodes.

Required Components

None

Back to top

Lumen_SetState

Node Type

Action Node

Description

Modifies the state of the Lumen Stealth Addon in the entity based on the specified new state (enabled, disabled, or changed). It retrieves the necessary component from the entity and invokes the appropriate method to set the new state.

Returned State

The node returns Failure if the Senses Asset is not detected or if the Stealth Component is not attached to the GameObject; otherwise, it returns Success.

Required Components

Senses Asset Pack, Stealth Component.

Back to top

Material_Change

Node Type

Action Node

Description

Modifies the materials of all child renderers within the entity based on specified types (Blackboard or Node) and field names. Materials can either be retrieved from the Blackboard using reflection or directly assigned from the nodes serialized field. The node applies the new materials to the corresponding renderers, ensuring the length of the materials array matches that of the renderers' material arrays.

Returned State

The node returns Failure if no renderers are found or if the materials array lengths don't match, otherwise, it returns Success.

Required Components

Renderers

Back to top

Material_Flash

Node Type

Action Node

Description

Temporarily changes the materials of all child renderers within the entity to a specified flash material, either obtained from the Blackboard or assigned directly from the node. It supports both constant and random durations for how long the flash material remains active. After the duration, the original materials are restored.

Returned State

The node returns Failure if no renderers are found or if materials cannot be retrieved from the Blackboard. Running is returned while material is changed. Success is returned when the flash period has ended and materials are restored.

Required Components

Renderers

Back to top

Nav_Chase

Node Type

Action Node

Description

Directs the AI agent to pursue a specified target using Unity's NavMesh system. The action updates the agent's path to the target's position, ensuring the agent is not stopped and handles path recalculation as necessary. It checks the agent's remaining distance to determine when the target is reached, stopping the agent if it is within the specified stopping distance.

Returned State

The node returns Failure if no target or agent is found, Success when the agent reaches the target, and Running while the agent is pursuing the target.

Required Components

NaveMesh Agent

Back to top

Nav_Flee

Node Type

Action Node

Description

Directs the agent to flee from a target by calculating a desired destination based on specified flee distance settings. The flee distance can either be constant or randomly chosen within a defined range. The agent sets its destination to a point in the opposite direction of the target if it is within the chosen flee distance.

Returned State

The node returns Failure if no target or agent is found, otherwise, it returns Success when the agent has moved beyond the stopping distance from the target, and returns Running if the agent is still fleeing.

Required Components

NaveMesh Agent

Back to top

Nav_MoveTowards

Node Type

Action Node

Description

Directs the agent to move towards a specified destination using a NavMesh path. The agent recalculates its path to the destination if the current destination changes and updates its movement accordingly. The node ensures that the agent continues moving until it is within the stopping distance of the destination.

Returned State

The node returns Failure if no agent or destination is specified; otherwise, it returns Success when the agent reaches the stopping distance, and Running while the agent is still in transit.

Required Components

NaveMesh Agent

Back to top

Nav_Patrol

Node Type

Action Node

Description

Directs the agent to patrol through a series of waypoints (Blackboard.Waypoints). The agent can either start at the nearest waypoint or the first in the array, depending on the specified start method. The patrol can run endlessly or stop after reaching the final waypoint. If running endlessly, the patrol can invert the order of waypoints upon completion.

Returned State

The node returns Failure if no agent or waypoints (Blackboard.Waypoints) are specified; Success when the patrol ends, and Running while the agent is moving towards the current waypoint.

Required Components

NaveMesh Agent

Back to top

Nav_ReachedDestination

Node Type

Branching Node

Description

Determines whether the agent has reached its destination within a specified accuracy, with an option to ignore height differences. If the condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children.

Returned State

This node returns Failure if the NavMesh Agent or destination is null; otherwise, it returns the update result of the child nodes.

Required Components

NavMesh Agent.

Back to top

Nav_Stop

Node Type

Action Node

Description

Stops the NavMesh Agent, ensuring it is no longer moving.

Returned State

If the agent is null, the node returns Failure. If the stop action is executed successfully, it returns Success.

Required Components

NaveMesh Agent

Back to top

Noise_Release

Node Type

Action Node

Description

Evaluates and applies noise strength based on various entry types and configurations, such as constant values, random ranges, and curves over time. The action retrieves the necessary noise component from the entity and invokes the appropriate method to set the noise strength.

Returned State

The node returns Failure if the Senses Asset is not detected or if the Noise component is not attached to the GameObject; otherwise, it returns Success.

Required Components

Senses Asset Pack, Noise Component.

Back to top

RandomChance

Node Type

Decorator Node

Description

Executes its child node only if a randomly generated value falls within a specified chance percentage.

Returned State

If the random condition is met, it returns the result of the child node; otherwise, it returns Failure. The node returns Failure if the child node fails to execute properly or if the chance condition is not met.

Required Components

None

Back to top

RandomDestination

Node Type

Action Node

Description

Selects a random position within a defined range (Min/Max positions) around the GameObject's current location. It generates a random offset from the current position and checks if the generated position is valid on the NavMesh surface if required. If the destination is valid, it updates the Blackboard.Destination with the new value.

Returned State

If the calculated destination is invalid (due to NavMesh validation), the node returns Failure; otherwise, it successfully updates Blackboard.Destination and returns Success.

Required Components

None

Back to top

Repeat

Node Type

Decorator Node

Description

Continuously executes its child node in a loop, maintaining a state of Running until explicitly stopped. It can be useful for tasks that need to be performed repeatedly until a specific condition is met or interrupted. The node's behavior continues indefinitely as long as it remains active, providing a way to create loops within the behavior tree structure.

Returned State

Returns Child State.

Required Components

None

Back to top

Retry

Node Type

Decorator Node

Description

Attempts to execute its child node up to a specified number of times until it succeed. It initializes the number of remaining attempts at the start and reduces this count with each failure. This node effectively allows for retrying actions that may fail due to transient conditions, ensuring that a task has multiple opportunities to succeed.

Returned State

If the child node returns a Failure state, it retries the execution until either a Success is achieved or the attempts run out.

Required Components

None

Back to top

RootNode

Node Type

Root Node

Description

Serves as the primary entry point for a behavior tree. It holds a single child node and executes its Update method during the update cycle. The RootNode manages its child's lifecycle but does not perform any additional logic.

Returned State

This node does not return a state by itself; it defers to the child node's update result.

Required Components

None

Back to top

Scent_Set

Node Type

Action Node

Description

Adjusts and updates the scent-related parameters like strength, visualization, gradient, and offset for a component using a senses integrator. Based on predefined conditions, it calculates and applies scent strength using either constant, random, or curve-based values over time. Updates various scent parameters such as strength, visualization, gradient, node separation, and layer.

Returned State

Return Failure if the required Senses Asset Pack is not installed or detected, else returns Success if the updates are applied correctly.

Required Components

Senses Asset Pack, Scent component.

Back to top

Scent_SetState

Node Type

Action Node

Description

Controls the enabled state of a scent component using a senses integrator. Based on the specified new state (enabled, disabled, or toggle), it adjusts the active state of the scent component. Modifies the enabled state of the scent component using the NewState condition (Enabled, Disabled, Change).

Returned State

Returns Failure if the Senses Asset Pack is not installed or the scent component is missing, otherwise returns Success.

Required Components

Senses Asset Pack, Scent component.

Back to top

ScentZoner_SetState

Node Type

Action Node

Description

Adjusts the state of a zoner scent addon using a senses integrator. Based on the specified state (enabled, disabled, or toggle), it either activates or deactivates the zoner scent functionality. Controls the zoner scent addon by enabling, disabling, or toggling its state.

Returned State

Returns Failure if the Senses Asset Pack is not installed or the scent component is missing, otherwise returns Success.

Required Components

Senses Asset Pack, Scent component.

Back to top

SeeBooster_Set

Node Type

Action Node

Description

Manages the visual enhancement buff for a component using the senses integrator. Based on the specified entry type (constant, random, time-based, or curve-based), it dynamically calculates and applies changes to the visual buff over time. This node adjusts the affected aspect and operation type of the buff and calculates the new buff value based on predefined curves or constants.

Returned State

It returns Failure if the Senses Asset Pack is not installed or the SeeBooster component is missing; otherwise, it returns Success if the updates are applied correctly.

Required Components

Senses Asset Pack, SeeBooster component.

Back to top

SeeBooster_SetState

Node Type

Action Node

Description

Controls the activation state of the visual enhancement component (SeeBooster) using the senses integrator. Based on the NewState (Enabled, Disabled, or Change), it updates the active state of the SeeBooster component, toggling its functionality in the scene.

Returned State

Returns Failure if the Senses Asset Pack is not installed or the SeeBooster component is missing. Returns Success if the state change is applied successfully.

Required Components

Senses Asset Pack, SeeBooster component.

Back to top

Selector

Node Type

Selector Node

Description

Manages the execution of child nodes, returning the state based on their outcomes.

Returned State

If a child returns Success, subsequent siblings are marked as Slipping. If a child returns Running, it halts execution and continues on the next update. If a child returns Failure, continue with next child check .If all children fail, it returns Failure.

Required Components

None.

Back to top

Senses_GetTarget

Node Type

Action Node

Description

Retrieves a target based on specified awareness and faction criteria. It initializes parameters upon start and invokes a detection method using the Senses component. If a target is detected, it updates the (Blackboard.Target) with the detected target's transform;

Returned State

Returns Failure if the Senses Asset Pack is not installed or the Senses component is missing.

Required Components

Senses Asset Pack, Senses component.

Back to top

Sequencer

Node Type

Composite Node

Description

Manages the execution of child nodes, returning the state based on their outcomes.

Returned State

Returning Running if a child is still active, Success if a child succeeds, and Failure if a child fails. Upon failure, subsequent children are marked as Slipping. If all children have executed, it returns Success; otherwise, it continues to return Running.

Required Components

None.

Back to top

Shoot

Node Type

Action Node

Description

Spawns a projectile (Prefab Rigidbody) and applies force to it in the direction of the target. The position, rotation, and scale of the spawned object can be configured using constant values or random values within specified ranges. The target, projectile, and necessary force are all required for this action to succeed.

Returned State

This node fails if no Rigidbody Prefab is assigned, or if Blackboard.Target == null.

Required Components

None.

Back to top

SphereCollider_Set

Node Type

Action Node

Description

Updates the size and radius of a specified SphereCollider based on the declared type (use Blackboard - for Scene Object or Node - for Prefabs). Allows for update based on various entry types, including constants, random ranges, and animation curves. :

Returned State

This node returns Failure if the SphereCollider is null.

Required Components

SphereCollider.

Back to top

Stealth_Set

Node Type

Action Node

Description

Adjusts the stealth buff of an agent by applying a value or curve over time. The new stealth buff value can be set directly, generated randomly between two constants, or based on an animation curve. Stealth-related values can change continuously depending on the selected entry type.

Returned State

Returns Failure if the Senses Asset Pack is not installed or the Stealth component is missing.

Required Components

Senses Asset Pack, Stealth component.

Back to top

Stealth_SetState

Node Type

Action Node

Description

Changes the enabled state of the stealth system for an agent based on the specified NewState. The node toggles the stealth component on or off, or switches between states if set to Change.

Returned State

Returns Failure if the Senses Asset Pack is not installed or the Stealth component is missing

Required Components

Senses Asset Pack, Stealth component.

Back to top

Succeeder

Node Type

Decorator Node

Description

Executes its child node and always returns Success, regardless of whether the child fails or succeeds. The child node will still execute and update its state, but this node forces a success outcome. This node is useful when you want to ensure that a particular branch of the tree does not cause a failure to propagate.

Returned State

Always return Success.

Required Components

None.

Back to top

Success

Node Type

Action Node

Description

Immediately returns Success without performing any operations or conditions. This node can be useful as a placeholder or when you need to guarantee a success outcome in a specific part of the tree.

Returned State

Always return Success.

Required Components

None.

Back to top

Target_Check

Node Type

Branching Node

Description

Checks if the Blackboard.Target is not null. If a target is available, it proceeds to execute the Left Output Children (or corresponding behavior). Otherwise, it runs the Right Output Children. This allows for conditional execution based on whether a target exists in the blackboard.

Returned State

This node returns Fail if the Blackboard.Target is null, otherwise it returns the update result of the child nodes.

Required Components

None

Back to top

Target_Check_2

Node Type

Action Node

Description

Checks whether Blackboard.Target is not null. If a valid target exists, the node succeeds; otherwise, it fails. This node provides a simple binary check for the presence of a target.

Returned State

This node returns Failure if Blackboard.Target is null, otherwise it returns Success.

Required Components

None.

Back to top

Target_Forget

Node Type

Action Node

Description

Resets Blackboard.Target to null, effectively clearing any previously assigned target. This action is useful for removing unwanted target references during behavior execution.

Returned State

This node always returns Success, as its operation is straightforward and does not depend on any conditions.

Required Components

None.

Back to top

Target_Search

Node Type

Action Node

Description

Searches for a target within a specified range using the defined criteria (either by tag or MonoBehaviour). It detects colliders within the range and checks if they match the specified conditions. If a matching target is found, it assigns the target to Blackboard.Target.

Returned State

This node returns Failure if the search range is zero, if searching by tag with no tags specified, or if searching by MonoBehaviour with no specified MonoBehaviour. If a matching target is found, it returns Success.

Required Components

None.

Back to top

Throw

Node Type

Action Node

Description

Instantiates a specified Rigidbody prefab and applies a force to it, throwing it towards a target. The prefab can be adjusted in rotation, position, and scale based on defined entry types. It uses a projectile calculation to determine the appropriate velocity to reach the target based on initial speed and gravity.

Returned State

This node returns Failure if the Rigidbody prefab is null, or if the target is null, else it returns Success.

Required Components

None.

Back to top

Timer

Node Type

Decorator Node

Description

Executes the child node for a specified amount of time. If the time limit is exceeded, it sets the child's state to 'Slipping' and returns 'Failure

Returned State

This node returns the child's state if time has not run out; otherwise, it returns 'Failure'.

Required Components

None.

Back to top

Tra_ChargeTowards

Node Type

Action Node

Description

Moves the controlled entity (Blackboard.This) towards the target's position at a specified speed until it reaches a defined stopping distance. Not rotating towards target during movement. It can override the blackboard's speed and stopping distance settings based on specified override types. If the target is null or the entity is outside the initial distance range, it will stop the movement.

Returned State

This node returns Failure if the target is null; returns Running while the entity is charging; and returns Success once the target is within the stopping distance.

Required Components

None.

Back to top

Tra_Chase

Node Type

Action Node

Description

Moves the controlled entity (Blackboard.This) towards the target (Blackboard.Target) while maintaining a defined stopping distance. Rotating towards target during movement. It allows for configurable speed and rotation settings, including an option for smooth rotation using Slerp.

Returned State

This node returns Failure if the target is null; it returns Running while the entity is moving towards the target; and returns Success once the entity is within the stopping distance.

Required Components

None.

Back to top

Tra_Flee

Node Type

Action Node

Description

Executes a flee behavior where the controlled entity (Blackboard.This) moves away from a target (Blackboard.Target). The distance to flee can be set as a constant or randomly determined between two specified values. It includes configurable parameters for speed, stopping distance, and rotation.

Returned State

If the target is null, it returns Failure. It returns Running while moving away and Success once the entity is beyond the stopping distance.

Required Components

None.

Back to top

Tra_LookAt

Node Type

Action Node

Description

Executes a look-at behavior where the controlled entity (Blackboard.This) rotates to face a specified target (Blackboard.Target). The target can be a Vector3, Transform, or GameObject, and height adjustments can be ignored. The rotation can be performed smoothly (slerp) or instantaneously.

Returned State

It returns Failure if the target is not valid, Running while interpolating the rotation, and Success when the entity is successfully facing the target.

Required Components

None.

Back to top

Tra_MoveTowards

Node Type

Action Node

Description

Implements a movement behavior that makes the controlled entity (Blackboard.This) move towards a specified destination (Blackboard.Destination) while optionally rotating to face it. The movement stops when the entity is within a defined stopping distance. The rotation can be smooth (using slerp) or immediate.

Returned State

It returns Failure if the destination is invalid, Running while the entity moves, and Success when the entity reaches the stopping distance.

Required Components

None.

Back to top

Tra_Patrol

Node Type

Action Node

Description

Implements a patrol behavior that allows an entity (Blackboard.This) to move between predefined waypoints (Blackboard.Waypoints). The entity can start at the nearest waypoint or the first waypoint in the array, and it can rotate towards each waypoint either smoothly (using slerp) or immediately. The patrol can be configured to run endlessly or stop after reaching the last waypoint. Additionally, it can invert the patrol order upon finishing.

Returned State

The state returns Failure if there are no waypoints, Running while moving between waypoints, and Success if the patrol stops.

Required Components

None.

Back to top

Tra_ReachedDestination

Node Type

Branching Node

Description

Checks if the distance between an entity (Blackboard.This) and a specified destination meets a defined accuracy threshold. It can ignore height differences for 2D checks. If the accuracy condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children.

Returned State

This node returns Failure if the specified destination in the Blackboard is null or if the accuracy condition is not met; otherwise, it returns the update result of the child nodes.

Required Components

None

Back to top

Transform_SetPosition

Node Type

Action Node

Description

Updates the position of a specified Transform based on various entry types, including constants, random ranges, and animation curves. This node allows scene objects to be referenced through the Blackboard, facilitating dynamic and flexible object manipulation. The Transform can be retrieved either directly from a specified reference or from the Blackboard based on the declared type.

Returned State

This node returns Failure if the specified Transform is null or if any other parameters are improperly set; otherwise, it returns Success after successfully updating the Transform's position.

Required Components

None.

Back to top

Transform_SetRotation

Node Type

Action Node

Description

Updates the rotation of a specified Transform based on various entry types, including constants, random ranges, and animation curves. This node enables scene objects to be referenced through the Blackboard, allowing for dynamic manipulation of rotation values. The target Transform can be either specified directly or retrieved from the Blackboard based on the declared type.

Returned State

This node returns Failure if the specified Transform is null or if any parameters are improperly set; otherwise, it returns Success after successfully updating the Transform's rotation.

Required Components

None.

Back to top

Transform_SetScale

Node Type

Action Node

Description

Updates the scale of a specified Transform based on various entry types, allowing for dynamic scaling behavior in the scene. This node references scene objects through the Blackboard, enabling flexible manipulation of scale values during runtime. It can set the Transform directly or retrieve it from the Blackboard according to the declared type.

Returned State

This node returns Failure if the specified Transform is null or if any parameters are improperly configured; otherwise, it returns Success after successfully applying the scale update.

Required Components

None.

Back to top

Variable_Check_A

Node Type

Action Node

Description

Evaluates a specified variable within the Blackboard against a required value using a defined comparison type. It determines whether the variable meets the specified condition, allowing for dynamic branching in behavior trees.

Returned State

This node returns Fail if the specified field in the Blackboard is null, otherwise, it returns the comparison condition result.

Required Components

None

Back to top

Variable_Check_B

Node Type

Branching Node

Description

Evaluates a specified variable within the Blackboard against a required value using a defined comparison type. It determines whether the variable meets the specified condition, allowing for dynamic branching in behavior trees. If the condition is met, it proceeds to execute the Left Output Children; otherwise, it runs the Right Output Children.

Returned State

This node returns Failure if the specified field in the Blackboard is null, the variable type does not match the expected type, or if the comparison condition is not met; otherwise, it returns the update result of the child nodes.

Required Components

None

Back to top

Variable_Set

Node Type

Action Node

Description

Modifies a specified variable within the Blackboard based on defined types and override conditions. It can set the variable to a new value, change it by a specified amount, or multiply it by a given factor. This allows for dynamic control over the state of variables within behavior trees. If the variable type does not match the expected type, the operation will fail.

Returned State

This node returns Failure if the operation fails; otherwise, it returns Success upon successfully updating the variable.

Required Components

None.

Back to top

Wait

Node Type

Action Node

Description

Pauses the execution of the behavior tree for a specified duration. It supports two modes, constant waiting for a fixed duration or random waiting for a duration between defined minimum and maximum values. The node returns Success once the waiting time has elapsed; otherwise, it returns Running while the wait is still in progress.

Returned State

This node returns Failure if the specified waiting time is invalid or uninitialized; otherwise, it returns Success when the wait time is completed, and Running while the waiting is ongoing.

Required Components

None.

Back to top

Zoner_SetState

Node Type

Action Node

Description

Modifies the state of the Zoner Stealth Addon functionality within the Senses Asset. The node can set the Zoner Stealt Addon state to Enabled, Disabled, or Change, toggling its current state. It checks for the presence of necessary components before executing and returns Success after modifying the state.

Returned State

This node returns Failure if the Senses Asset is not installed or if the Stealth component is missing; otherwise, it returns Success after successfully updating the zoner state.

Required Components

Senses Asset Pack, Stealth component.

Back to top