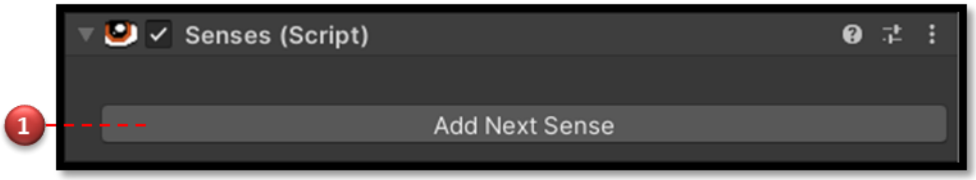
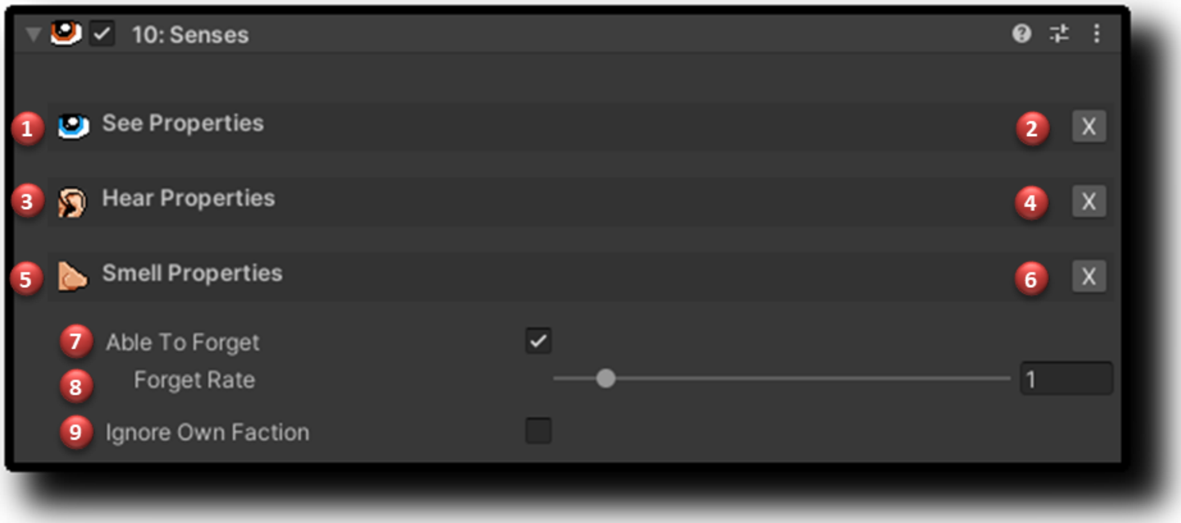
Senses Component

Description

Allows for Managing Sensors of character (Adding, Removing, Configuring) in order to gather (and further proceed) Awareness.



1	Add Next Sense Button	Allows adding next sense sensors.



Able To Forget ☑
Forget Rate 1
Ignore Own Faction □

1	See Properties Foldout	Allows showing / hiding of See Sensor properties.
2	Remove Button	Allows disabling See Sensor.
3	See Properties Foldout	Allows showing / hiding of Hear Sensor properties.
4	Remove Button	Allows disabling Hear Sensor.
5	Smell Properties Foldout	Allows showing / hiding of Smell Sensor properties.
6	Remove Button	Allows disabling Smell Sensor.
7	Able To Forget	Able To Forget allows the setting of the Use_Forgetting state. If enabled, Awareness will decrease by the Forget Rate during each Update.
8	Forget Rate	Allows setting up float value of Awareness decreased during each time interval.

9	Ignore Own Faction	When Enabled will ignore TargetSenses with same Faction selected. If do not had Faction selected, will ignore TargetSenses with Faction set as None

Awareness

This floating-point value represents how well a character understands that something has been detected. It ranges from 0 to 100. Awareness is individually tracked for each **GameObject** equipped with the **TargetSenses** component.

Awareness had separate values for each enabled Sense Sensor. And each of this awareness had value in range from 0 to 100.

Forgetting - if **Able To Forget** is enabled, then awareness will be decreased by the **Forget**, according to bellow rules:

1	See Awareness	Will be reduces over time if a GameObject with the attached TargetSenses was not detected during the last check. This mechanism ensures that characters only forget about a target if it becomes no longer visible.
2	Hear Awareness	Will be reduces over time.
3	Smell Awarness	Will be reduces over time if a ScentNode released by a Scent attached to a GamoObject with TargetSenses was not detected during the last check. This mechanism ensures that characters only forget about a target if it is no longer smelled.

Public Properties

Use_See	Returns true if uses See Sensor.
Use_Hear	Returns true if uses Hear Sensor.
Use_Smell	Returns true if uses Smell Sensor.
Eyes	Get transform Eyes.
IgnoreLayers	Get Ignored Layers
Use_Forgeting	Returns true if uses forgetting.
Forgeting_Rate	Get float value of Forgeting_Rate
Use_IgnoreOwnFaction	Returns true if uses ignore own nation
Central_VisionAngle	Get float value of Central_VisionAngle
Central_VisionRadius	Get float value of Central_VisionRadius
Use Peripheral FOV	Returns true if uses peripheral field of view.
Peripheral_VisionAngle	Get float value of Peripheral_VisionAngle
Peripheral_VisionRadius	Get float value of Peripheral_VisionRadius
Use_DisplayFOV	Returns true if uses display field of view.
Use_IgnoreRotationX	Returns true if uses ignore rotation in X axis.
Use_RayCastInFovOnly	Returns true if uses raycast in field of view only.
Use_DebugDrawRay	Returns true if uses debug draw ray.
Use_OffsetRotation	Returns true if uses rotation offset.
Offset_Rotation	Get Vector3 value of rotation offset.

Use_OffsetPosition	Returns true if uses position offset.
Offset_Position	Get Vector3 value of position offset.
Hear_Sensitivity	Returns enum value of Hear_Sensitivity.
Smell_Radius	Get float value of Smell Radius
ScentLayer	Get Layer dedicated for scents.
DidSee	Allows for checking did chosen TargetSenses was Seen
DidHear	Allows for checking did chosen TargetSenses was Heard
DidSmell	Allows for checking did chosen TargetSenses was Smelled
WasAnythingSeen	Allows for checking did any TargetSenses was Seen
WasAnythingHeard	Allows for checking did any TargetSenses was Heard
WasAnythingSmelled	Allows for checking did any TargetSenses was Smelled
GetDetectedTargetList	Provides List of TargetSenses above required Awareness value.
GetRememberedTargetList	Provides List of TargetSenses below required Awareness value but above zero.
GetAllTargetList	Provides List of all ever detected TargetSenses.
ResetAwarness	Clears all gathered awareness data, without releasing memory. Recommended for usage in case of object pooling.

Public Methods

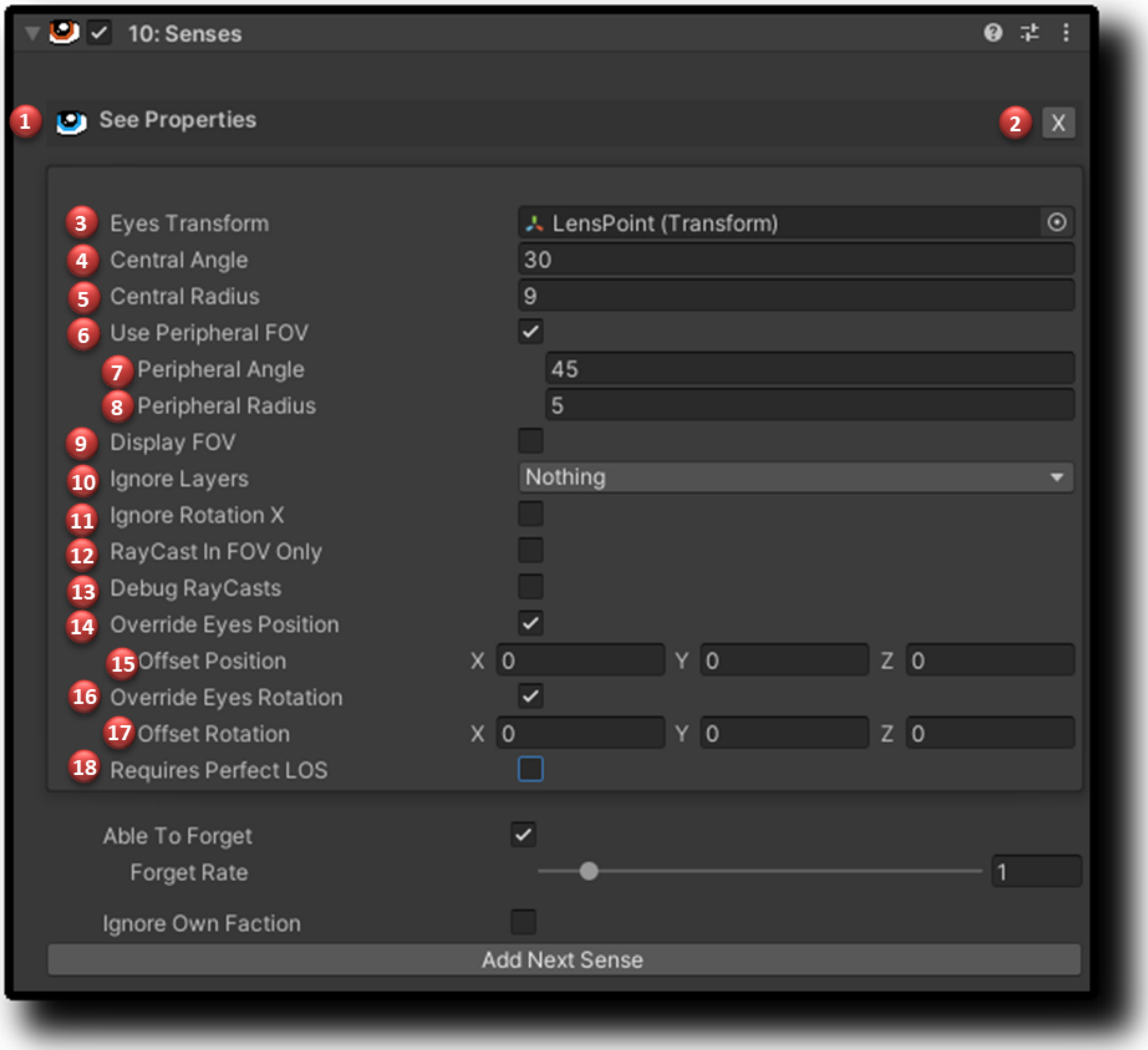
Set_Eyes	Sets transform Eyes.
Set_UseSee	Sets state of bool Use_See.
Set_UseHear	Sets state of bool Use_Hear.
Set_UseSmell	Sets state of bool Use_Smell.
Set_UseForgeting	Sets state of bool Use_Forgeting.
Set_ForgetingRate	Sets float value of Forgeting_Rate.
Set_UseIgnoreOwnFaction	Sets state of bool Use_IgnoreOwnFaction.
Set_CentralVisionAngle	Sets float value of Central_VisionAngle.
Set_CentralVisionRadius	Sets float value of Central_VisionRadius.
Set_UsePeripheralFOV	Sets state of bool Use_PeripheralFOV.
Set_PeripheralVisionAngle	Sets float value of Peripheral_VisionAngle.
Set_PeripheralVisionRadius	Sets float value of Peripheral_VisionRadius.
Set_UseDisplayFOV	Sets state of bool Use_UseDisplayFOV.
Set_UseIgnoreRotationX	Sets state of bool Use_IgnoreRotationX.
Set_UseRayCastInFovOnly	Sets state of bool Use_RayCastInFovOnly.
Set_DebugDrawRay	Sets state of bool Use_DebugDrawRay.
Set_UseOffsetRotation	Sets state of bool Use_OffsetRotation.
Set_OffsetRotation	Sets Vector3 value of OffsetRotation;
Set_UseOffsetPosition	Sets state of bool Use_OffsetPosition.
Set_OffsetPosition	Sets Vector3 value of OffsetPosition;
Set_HearSensitivity	Sets enum value of Hear_Sensitivity.
Recived_Noise	Increase value of Hear Awareness of provided TargetSenses
Set_SmellRadius	Sets float value of Smell_Radius.

See - Senses Sensor

Description

See is Senses Sensor allows for Observing objects in two complementary zones (Central Vision and Peripheral).

See Sensor could be enabled or disabled either through inspector tab or code.
See Sensor gather **See Awareness** of **GameObject** with attached **TargetSenses** component.
Properties of See Sensor could be setup either through inspector tab or code.



☑ 10: Senses

1 See Properties 2 X

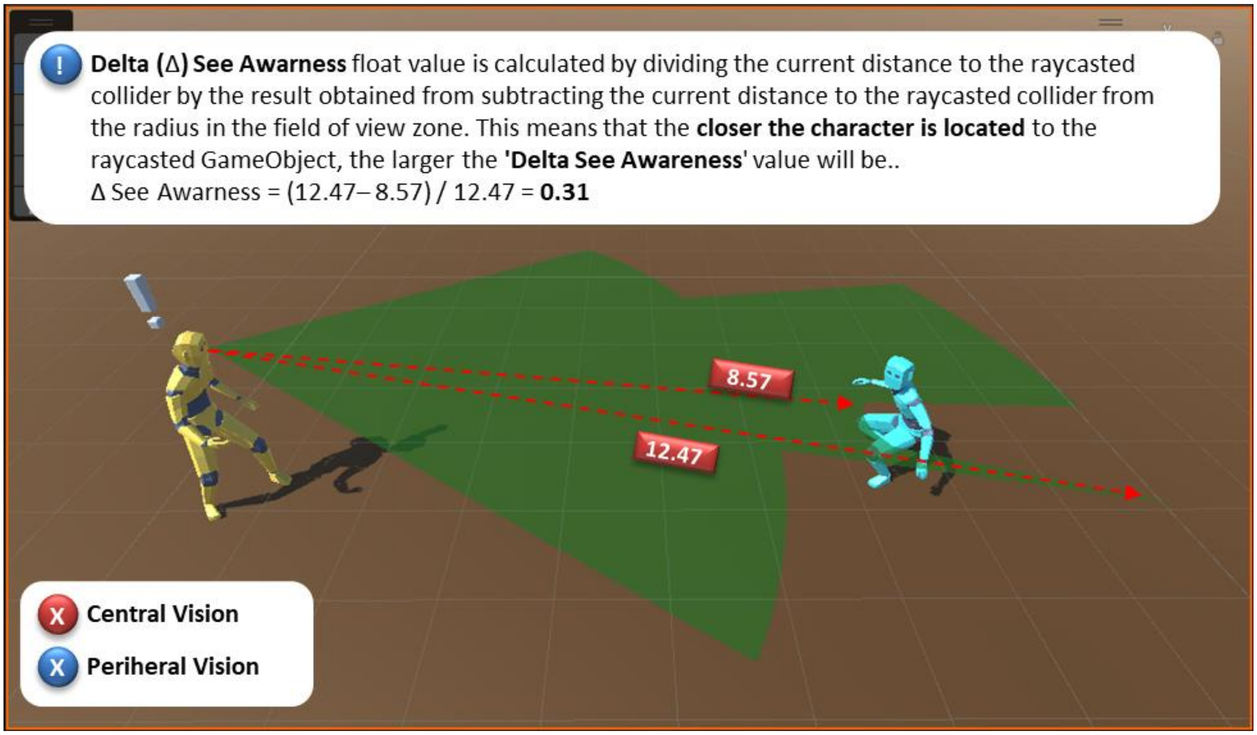
3 Eyes Transform: LensPoint (Transform)
4 Central Angle: 30
5 Central Radius: 9
6 Use Peripheral FOV: ☑
7 Peripheral Angle: 45
8 Peripheral Radius: 5
9 Display FOV: □
10 Ignore Layers: Nothing
11 Ignore Rotation X: □
12 RayCast In FOV Only: □
13 Debug RayCasts: □
14 Override Eyes Position: ☑
15 Offset Position: X 0 Y 0 Z 0
16 Override Eyes Rotation: ☑
17 Offset Rotation: X 0 Y 0 Z 0
18 Requires Perfect LOS: □

Able To Forget: ☑
Forget Rate: 1

Ignore Own Faction: □

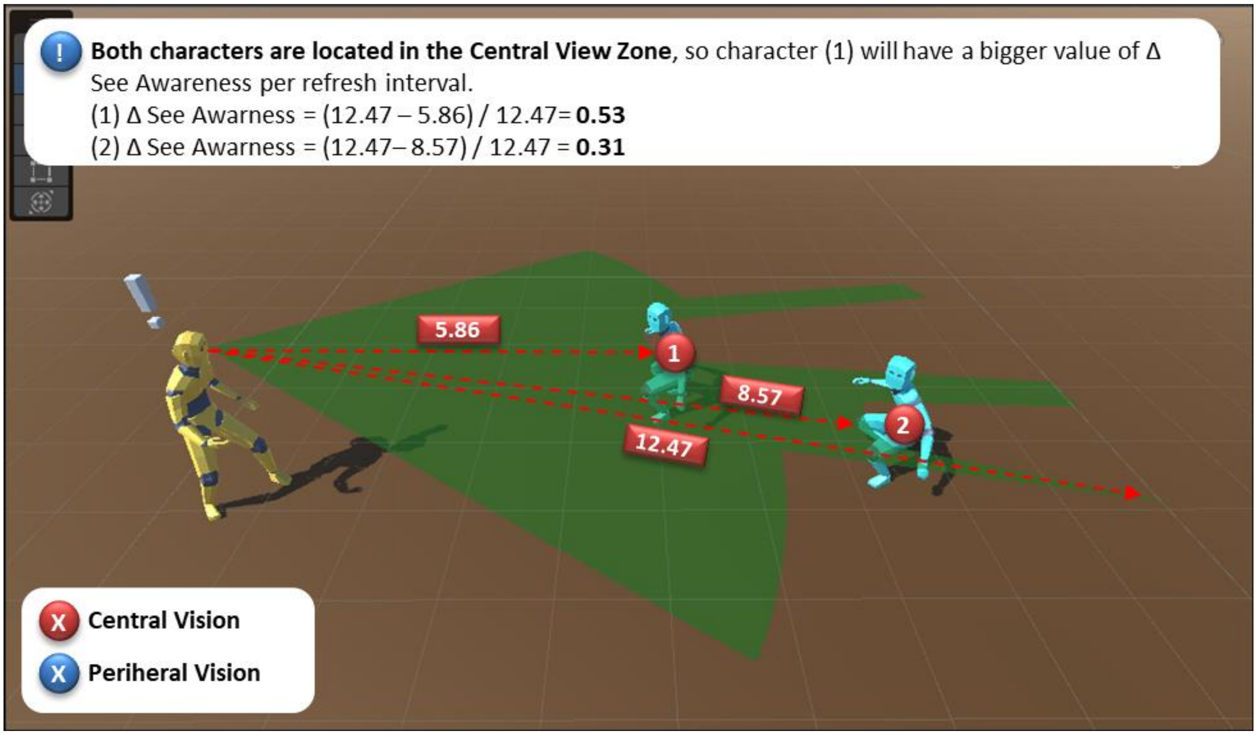
1	See Properties Foldout	Allows showing / hiding of See Sensor properties.
2	Remove Button	Allows disabling See Sensor.
3	Eyes Transform	Allows assign of Transform used as Eyes Transform. Eyes Transform position is used during radius, and angle calculations. Furthermore serves as raycast origin point during Line of Sight check. If not assign, script will use transform position.
4	Central Angle float	Allows setting up float value of central vision angle.
5	Central Radius float	Allows setting up float value of central vision radius.
6	Use Peripheral FOV	Allows setting state of use peripheral field of view.
7	Peripheral Angle float	Allows setting up float value of peripheral vision angle.
8	Peripheral Radius float	Allows setting up float value of peripheral vision radius.
9	Display FOV	Allows setting state of displaying field of view. Allowing for graphical preview of central and peripheral vision zone.
10	Ignore Layers	Allows setting up ignored layers.
11	Ignore Rotation X	Allows setting state of ignore rotation x. If enabled field of view will not rotate in X axis, and stay parallel to the ground level.
12	RayCast in FOV Only	When activated, the field of view will be treated as a flat triangle, causing colliders positioned below or above the field of view to be excluded from raycasting.
13	Debug RayCasts	When enabled, this feature allows for debugging raycasts. A green line indicates rays that did not hit anything, while a red line indicates rays that successfully hit an object.
14	Override Eyes Position	When enabled, Position Offset value will be applied to Eyes Transform (if Eyes Transform equals null, offset will be applied to transform.position)
15	Offset Position	Vector3 value affecting Eyes Transform (if Eyes Transform equals null, offset will be applied to transform.position). Offset Postion is only applied with enabled Override Eyes Position.
16	Override Eyes Rotation	When enabled, Position Rotation value will be applied to Eyes Transform (if Eyes Transform equals null, offset will be applied to transform.rotation)
17	Offset Rotation	Vector3 value affecting Eyes Transform (if Eyes Transform equals null, offset will be applied to transform.rotation). Offset Rotation is only applied with enabled Override Eyes Position.
18	Requires Perfect LOS	When Enabled, while raycasting, the Line of Sight will be blocked by any Collider attached to the TargetSenses component. When Disabled, the raycast will pass through the Collider attached to the TargetSenses component.

Delta See Awareness



Delta (Δ) See Awarness float value is calculated by dividing the current distance to the raycasted collider by the result obtained from subtracting the current distance to the raycasted collider from the radius in the field of view zone. This means that the **closer the character is located** to the raycasted GameObject, the larger the **'Delta See Awareness'** value will be..

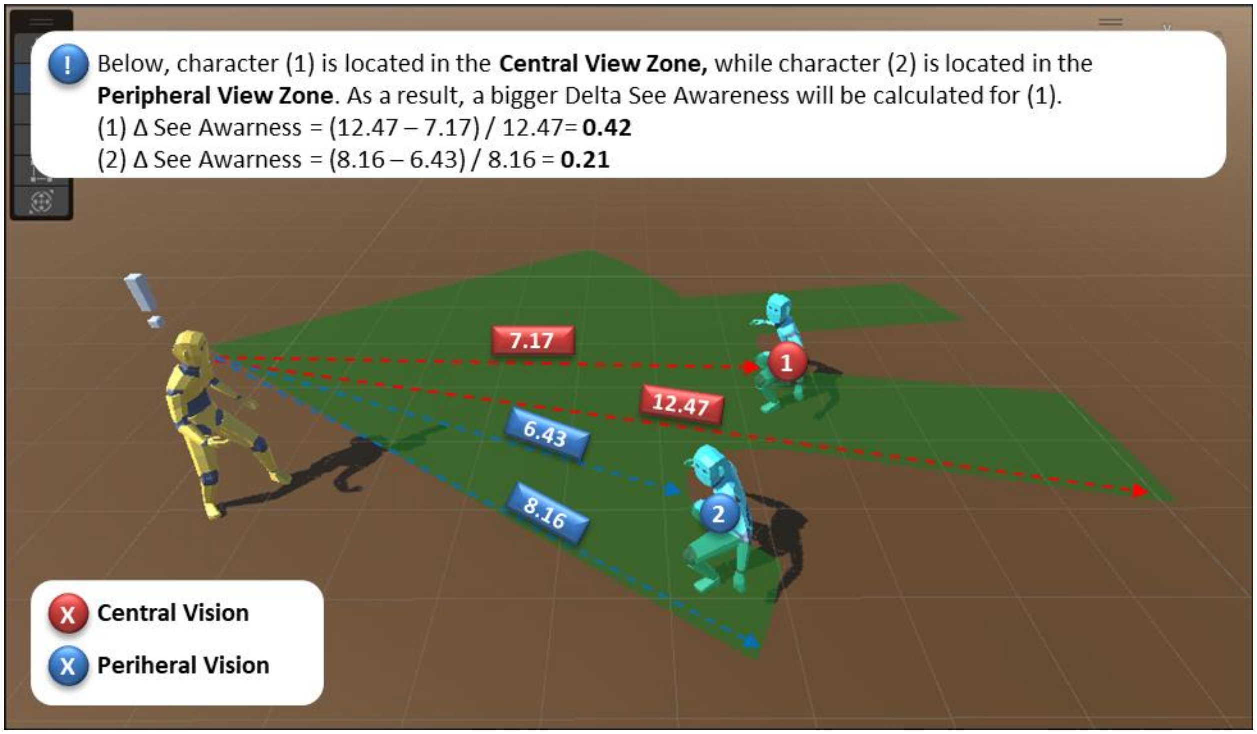
Δ See Awarness = (12.47– 8.57) / 12.47 = **0.31**



Both characters are located in the Central View Zone, so character (1) will have a bigger value of Δ See Awareness per refresh interval.

(1) Δ See Awarness = (12.47 – 5.86) / 12.47= **0.53**

(2) Δ See Awarness = (12.47– 8.57) / 12.47 = **0.31**

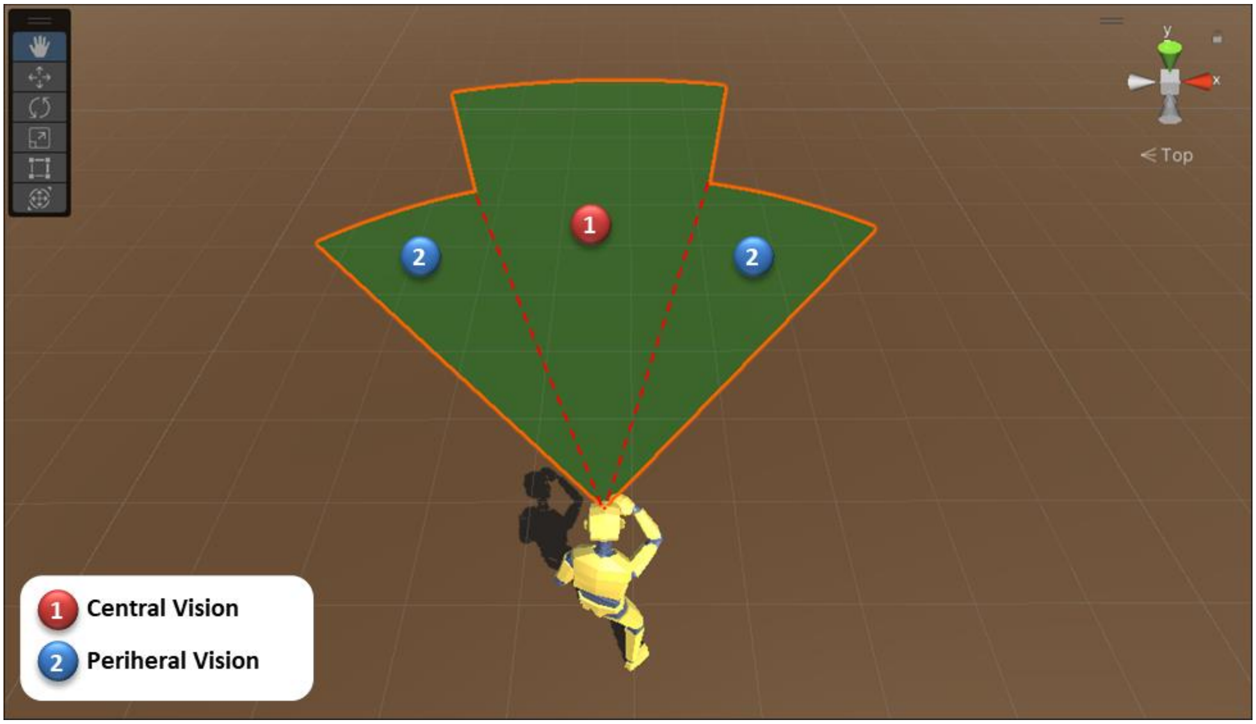


Below, character (1) is located in the **Central View Zone,** while character (2) is located in the **Peripheral View Zone**. As a result, a bigger Delta See Awareness will be calculated for (1).

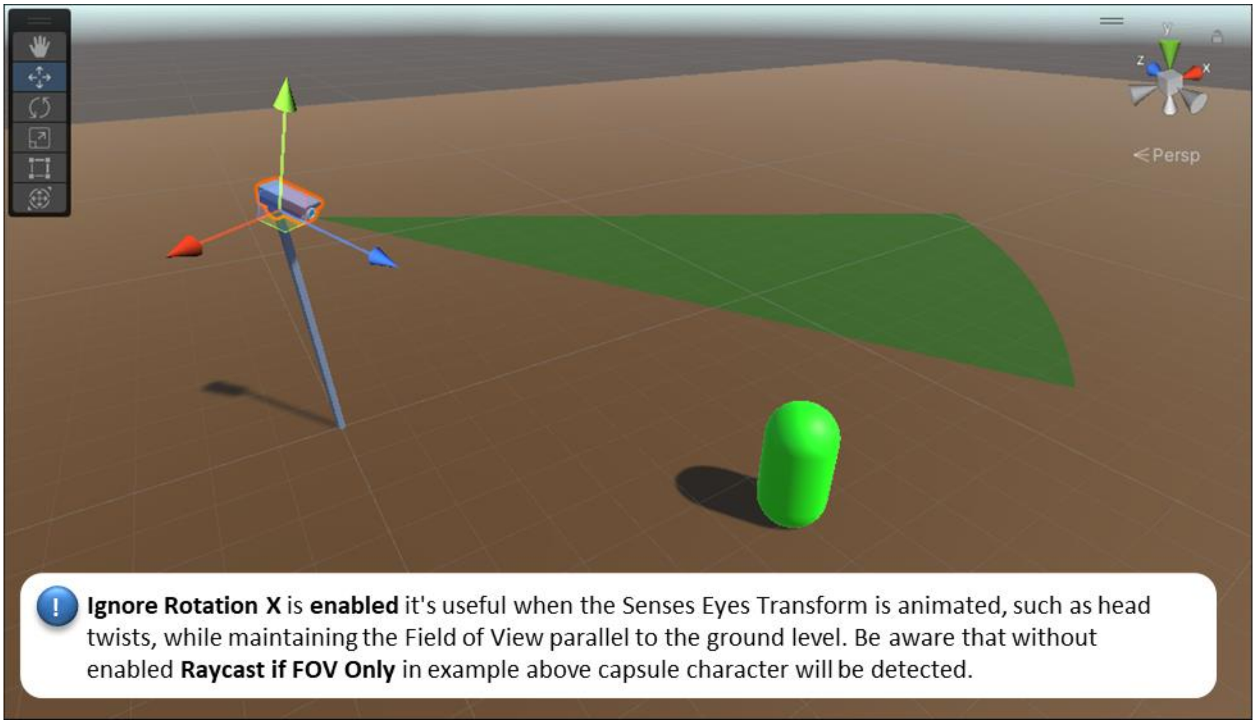
(1) Δ See Awarness = (12.47 – 7.17) / 12.47= **0.42**

(2) Δ See Awarness = (8.16 – 6.43) / 8.16 = **0.21**

Field of View Zones

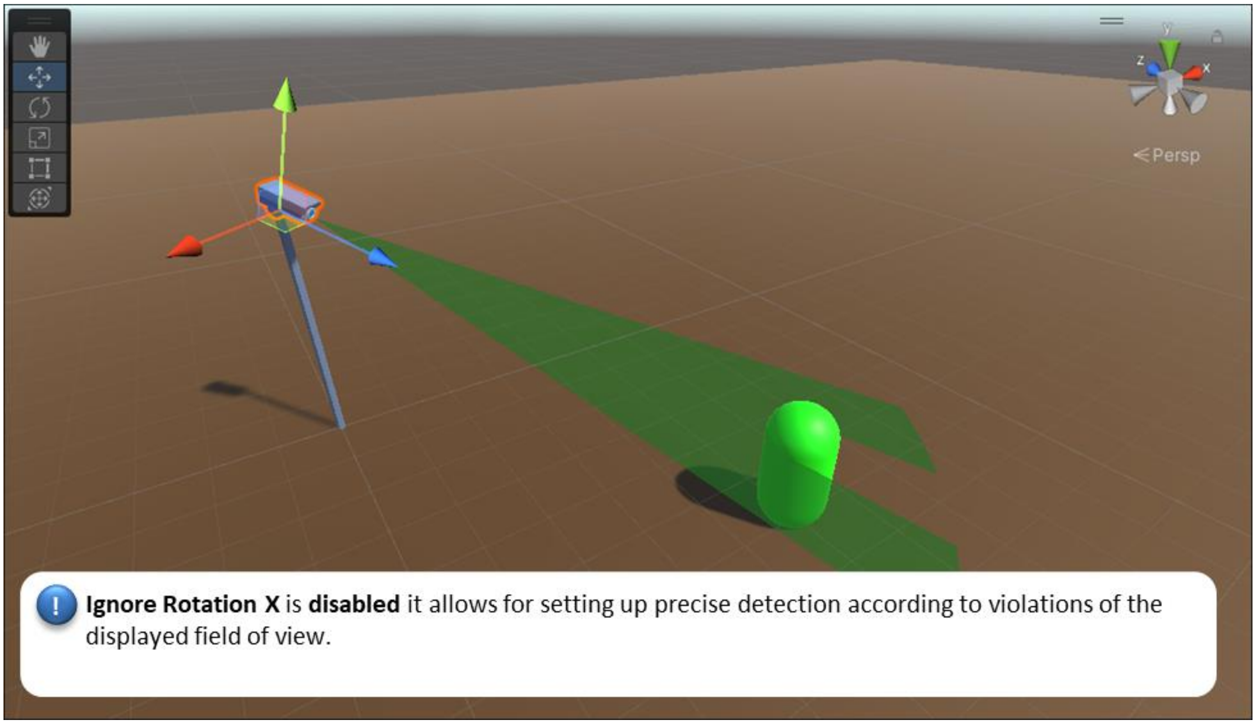


Ignore Rotation X (enabled).



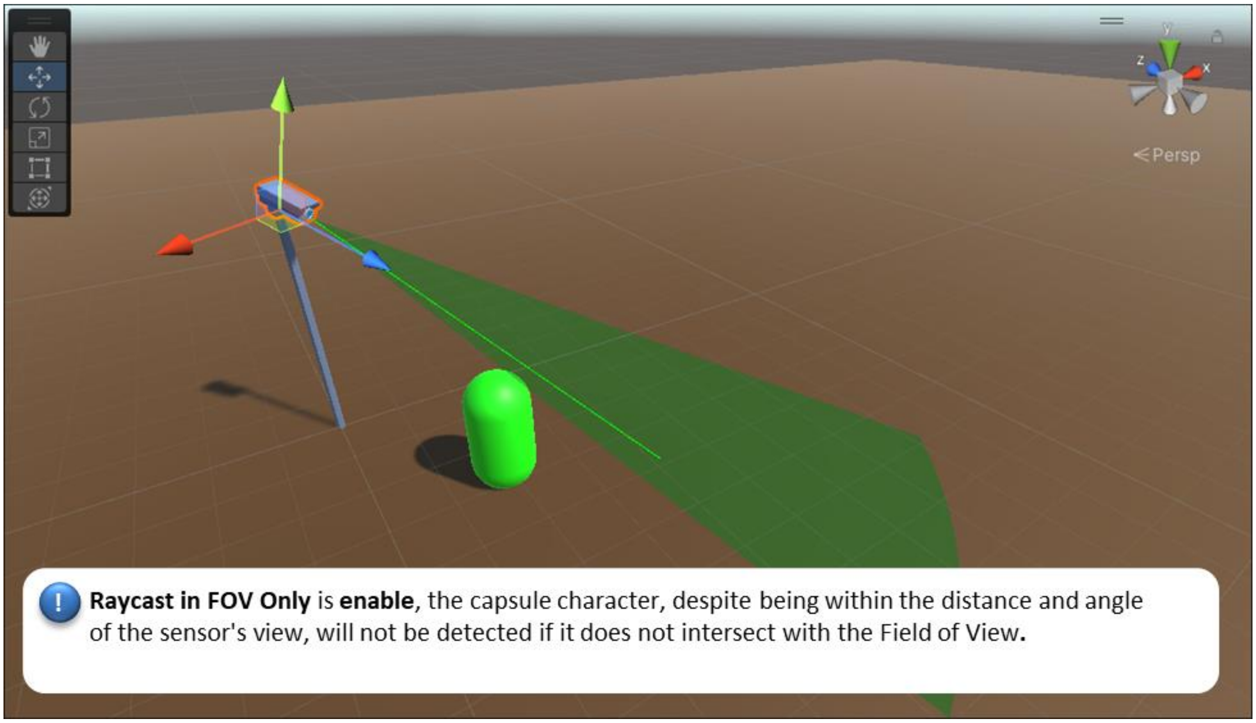
Ignore Rotation X is **enabled** it's useful when the Senses Eyes Transform is animated, such as head twists, while maintaining the Field of View parallel to the ground level. Be aware that without enabled **Raycast if FOV Only** in example above capsule character will be detected.

Ignore Rotation X (disabled).



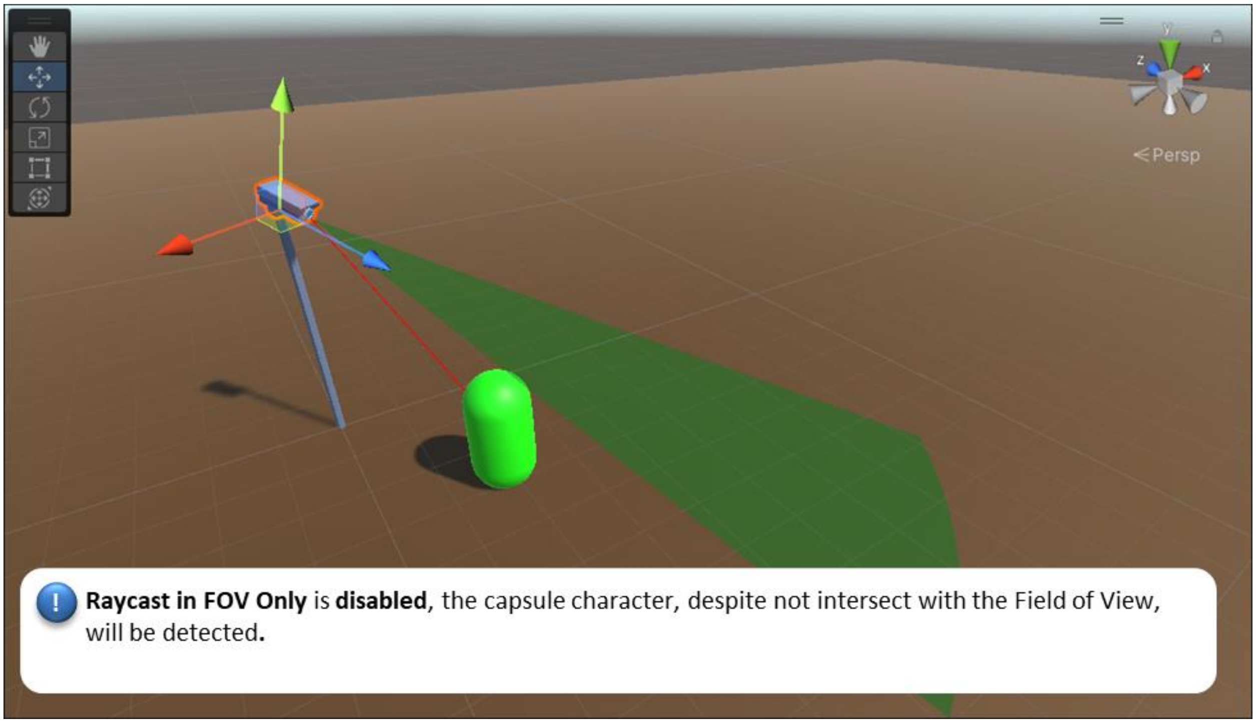
Ignore Rotation X is **disabled** it allows for setting up precise detection according to violations of the displayed field of view.

Raycast in FOV Only (enabled).



Raycast in FOV Only is **enable**, the capsule character, despite being within the distance and angle of the sensor's view, will not be detected if it does not intersect with the Field of View**.**

Raycast in FOV Only (disabled).



Raycast in FOV Only is **disabled**, the capsule character, despite not intersect with the Field of View, will be detected**.**

Public Properties

Use_See	Returns true if uses See Sensor.
Eyes	Get transform Eyes.
IgnoreLayers	Get Ignored Layers
Central_VisionAngle	Get float value of Central_VisionAngle
Central_VisionRadius	Get float value of Central_VisionRadius
Use_PeripheralFOV	Returns true if uses peripheral field of view.
Peripheral_VisionAngle	Get float value of Peripheral_VisionAngle
Peripheral_VisionRadius	Get float value of Peripheral_VisionRadius
Use_DisplayFOV	Returns true if uses display field of view.
Use_IgnoreRotationX	Returns true if uses ignore rotation in X axis.
Use_RayCastInFovOnly	Returns true if uses raycast in field of view only.
Use_DebugDrawRay	Returns true if uses debug draw ray.
Use_OffsetRotation	Returns true if uses rotation offset.
Offset_Rotation	Get Vector3 value of rotation offset.
Use_OffsetPosition	Returns true if uses position offset.
Offset_Position	Get Vector3 value of postiion offset.

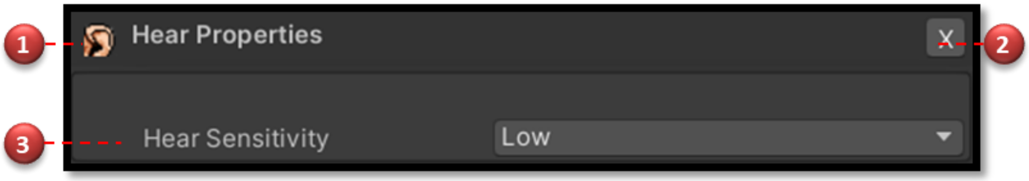
Public Methods

Set_UseSee	Sets state of bool Use_See.
Set_Eyes	Sets transform Eyes.
Set_CentralVisionAngle	Sets float value of Central_VisionAngle.
Set_CentralVisionRadius	Sets float value of Central_VisionRadius.
Set_UsePeripheralFOV	Sets state of bool Use_PeripheralFOV.
Set_PeripheralVisionAngle	Sets float value of Peripheral_VisionAngle.
Set_PeripheralVisionRadius	Sets float value of Peripheral_VisionRadius.
Set_UseDisplayFOV	Sets state of bool Use_UseDisplayFOV.
Set_UseIgnoreRotationX	Sets state of bool Use_IgnoreRotationX.
Set_UseRayCastInFovOnly	Sets state of bool Use_RayCastInFovOnly.
Set_DebugDrawRay	Sets state of bool Use_DebugDrawRay.
Set_UseOffsetRotation	Sets state of bool Use_OffsetRotation.
Set_OffsetRotation	Sets Vector3 value of OffsetRotation;
Set_UseOffsetPosition	Sets state of bool Use_OffsetPosition.
Set_OffsetPosition	Sets Vector3 value of OffsetPosition;

Hear - Senses Sensor

Description

Hear is Senses Sensor allows for hearing noises released by objects through Noise Component.
Hear Sensor could be enabled or disabled either through inspector tab or code.
Hear Sensor gather **Hear Awareness** of **GameObject** with attached **TargetSenses** component.
Properties of Hear Sensor could be setup either through inspector tab or code.



1	Hear Properties Foldout	Allows showing / hiding of Hear Sensor properties.
2	Remove Button	Allows disabling Hear Sensor.
3	Hear Sensitivity	Allows setting Hear_Sensitivity.

Public Properties

Use_Hear	Returns true if uses Hear Sensor.
Hear_Sensitivity	Returns enum value of Hear_Sensitivity.

Public Methods

Set_UseHear	Sets state of bool Use_Hear.
Set_HearSensitivity	Sets enum value of Hear_Sensitivity.
Recived_Noise	Increase value of Hear Awareness of provided TargetSenses

Smell - Senses Sensor

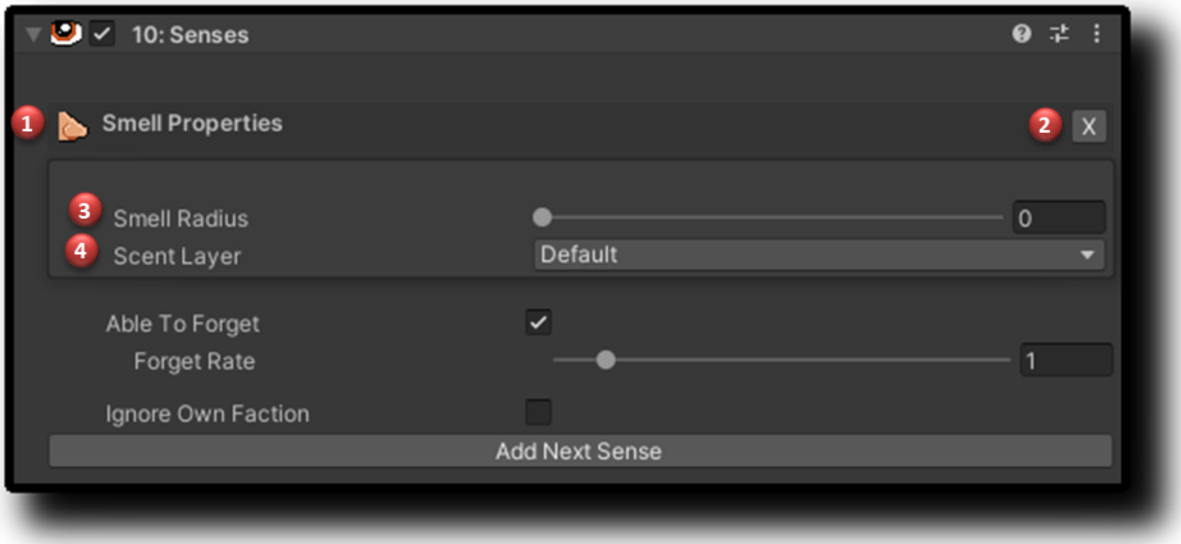
Description

Smell is Senses Sensor allows for detecting trail released by Scent Component and collecting awareness from it.

Smell Sensor could be enabled or disabled either through inspector tab or code.

Smell Sensor gather **Smell Awareness** of **GameObject** with attached **Scent** and **TargetSenses** component.

Properties of Smell Sensor could be setup either through inspector tab or code.



Smell Radius 0

Scent Layer Default

Able To Forget ☑

Forget Rate 1

Ignore Own Faction ☐

1	Smell Properties Foldout	Allows showing / hiding of Smell Sensor properties.
2	Remove Button	Allows disabling Smell Sensor.
3	Smell Radius	Allows setting Smell Radius
4	Scent Layer	Allows setting up scent dedicated Layer.

Public Properties

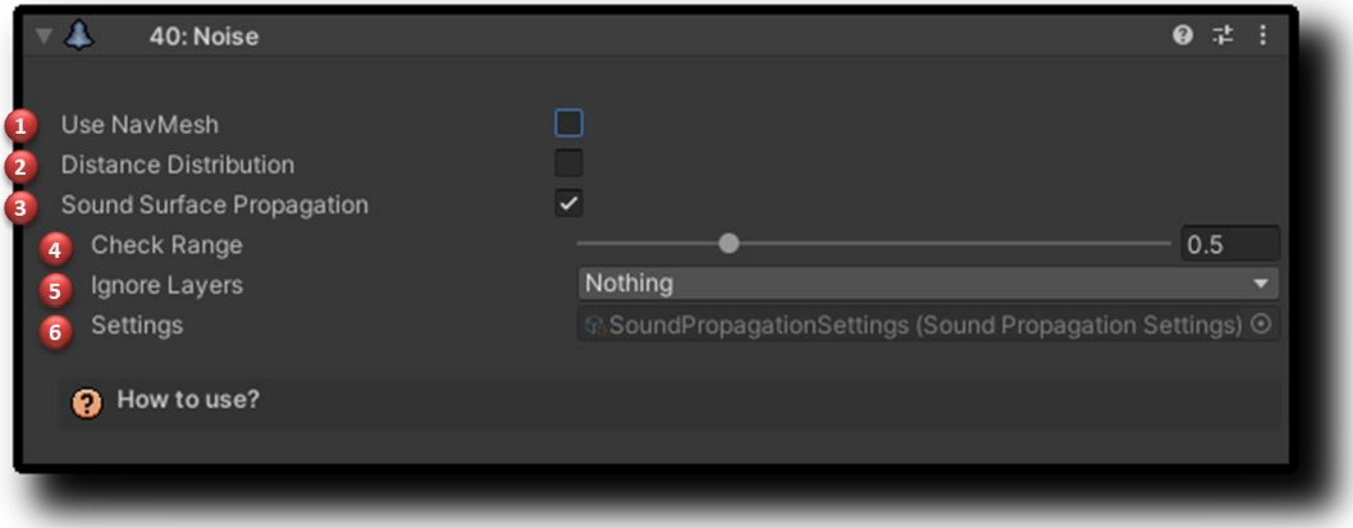
Use_Smell	Returns true if uses Smell Sensor.
Smell_Radius	Get float value of Smell Radius
ScentLayer	Get Layer dedicated for scents.

Public Methods

Set_UseSmell	Sets state of bool Use_Smell.
Set_SmellRadius	Sets float value of Smell_Radius.

Noise Component

Description



40: Noise

1 Use NavMesh ☐

2 Distance Distribution ☐

3 Sound Surface Propagation ☑

4 Check Range 0.5

5 Ignore Layers Nothing

6 Settings SoundPropagationSettings (Sound Propagation Settings)

How to use?

1	Use NaveMesh	When enabled, the distance to a Senses Component with a Hear Sensor will be calculated using the NavMesh. If no valid path is found, the distance will be calculated in a straight line.
2	Distance Distribution	When enabled, all Senses Components will receive emitted noise in order of distance (physically accurate solution). When disabled, all Senses Components will receive emitted noise in sphere cast order (performance-efficient solution).
3	Sound Surface Propagation	When enabled, the noise value will be multiplied based on the type of surface below the agent.
4	Check Range	Allows configuring the collider check range that affects sound propagation. If multiple elements are detected, sound propagation will be influenced by the element with the highest Y position.
5	Ignore Layers	Allows setting up ignored layers.
6	Settings	Allows setting up sound propagation settings, TAGs and Factors.

Properties

Use_NaveMesh	Returns value of Use_NaveMesh.
Use_SpreadAccordingToDistance	Returns value of Use_SpreadAccordingToDistance.

Public Methods

Set_UseNaveMesh	Sets value of Use_NaveMesh.
Set_UseSpreadAccordingToDistance	Sets value of Use_SpreadAccordingToDistance.
Release_Noise	When called spreading value of Noise among all

	characters with enabled Hear Senses Sensor.

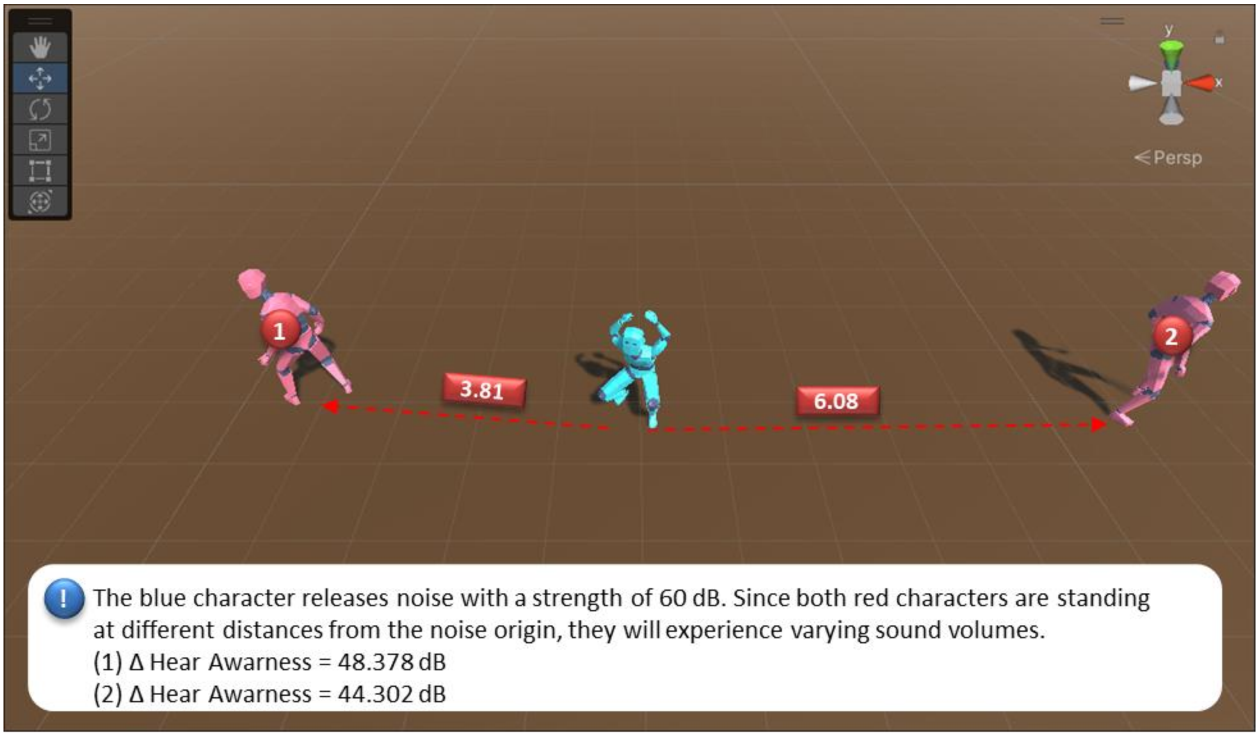
void Release_Noise (float _noiseValue)

When invoked, the noise value (_noiseValue) will be recalculated for all characters with the Hear Senses Sensor enabled, taking into account their distance from the originating NoiseComponent. If deemed significant, the recalculated noise value will then be distributed among characters capable of hearing.

The _noiseValue parameter represents the real-world sound power in decibels (dB). Example values are provided for reference.

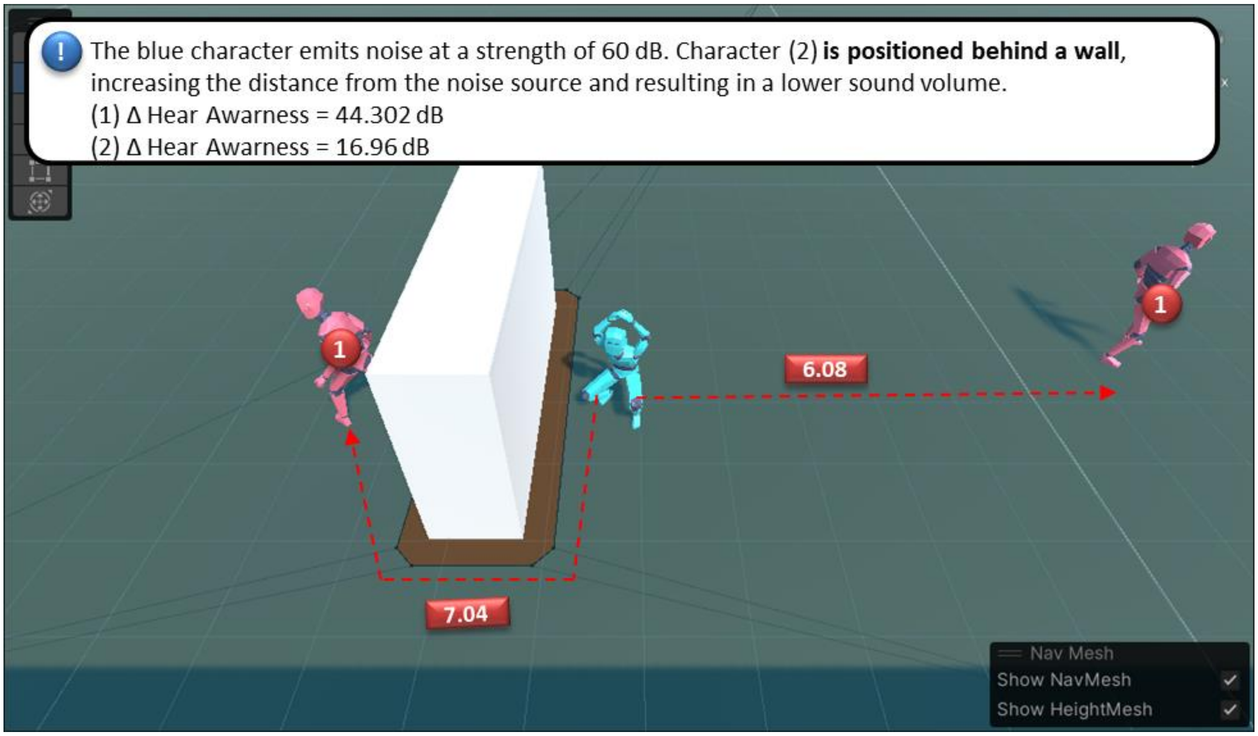
Footsteps	60 dB
Gunshot	140 dB
Explosion	160 dB
Wind	40 dB
Rain	50 dB
Fire crackling	70 dB
Car engine	80 dB
Bird chirping	70 dB
Water splash	90 dB
Crowd cheering	90 dB

Recalculated Noise Value



The blue character releases noise with a strength of 60 dB. Since both red characters are standing at different distances from the noise origin, they will experience varying sound volumes.
(1) Δ Hear Awarness = 48.378 dB
(2) Δ Hear Awarness = 44.302 dB

Use_NaveMesh



The blue character emits noise at a strength of 60 dB. Character (2) **is positioned behind a wall**, increasing the distance from the noise source and resulting in a lower sound volume.
(1) Δ Hear Awarness = 44.302 dB
(2) Δ Hear Awarness = 16.96 dB

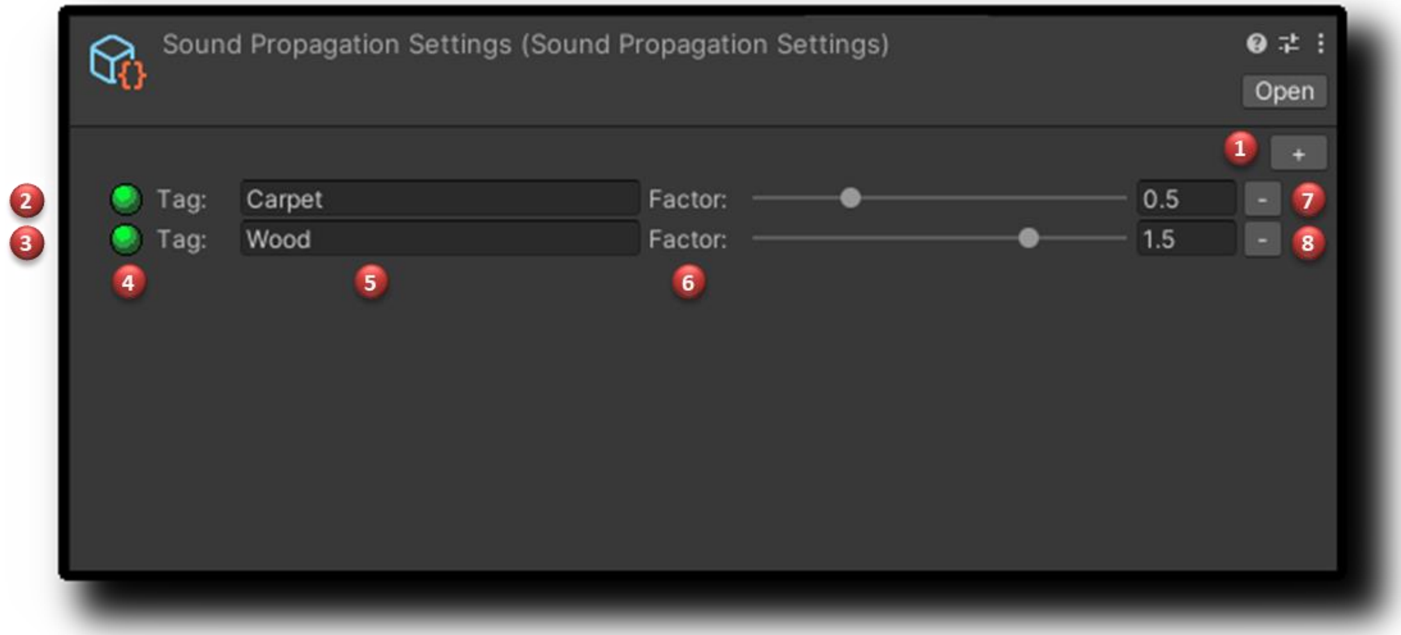
Sound Propagation

When the **Noise Component** has "Sound Surface Propagation" enabled, the Release_Noise() function performs a **sphere cast** to detect colliders within the defined **Check Range** that can influence **Sound Propagation**.

Propagation Mechanics:

- If any detected collider is included in **Sound Propagation Settings**, the _noiseValue will be **multiplied** by the associated propagation factor.
- If **multiple colliders** are detected, the **factor from the collider with the highest** Transform.position.y will be applied.

Sound Propagation Settings


Sound Propagation Settings (Sound Propagation Settings)

Tag: Carpet Factor: 0.5

Tag: Wood Factor: 1.5

1	Add Next Sound Propagation Data	Allows adding next Sound Propagation Data
2	Sound Propagation Data	Sound Propagation Data, for GameObject with TAG: “Carpet” with factor 0.5f;
3	Sound Propagation Data	Sound Propagation Data, for GameObject with TAG: “Wood” with

		factor 1.5f;
4	TAG check	Icon color inform about TAG status. Green mean TAG is correct, red mean TAG do not exist.
5	Text field	Allows setting up TAG string.
6	Factor	Allows setting up float value of factor which will multiply “_noiseValue”.
7,8	Remove Sound Propagation Data	Allows removing Sound Propagation Data.

Obstacle Component

Description

Mitigates the detection of objects located behind it.

Properties

Range_DecreaseFactor	Get float value of Range_DecreaseFactor

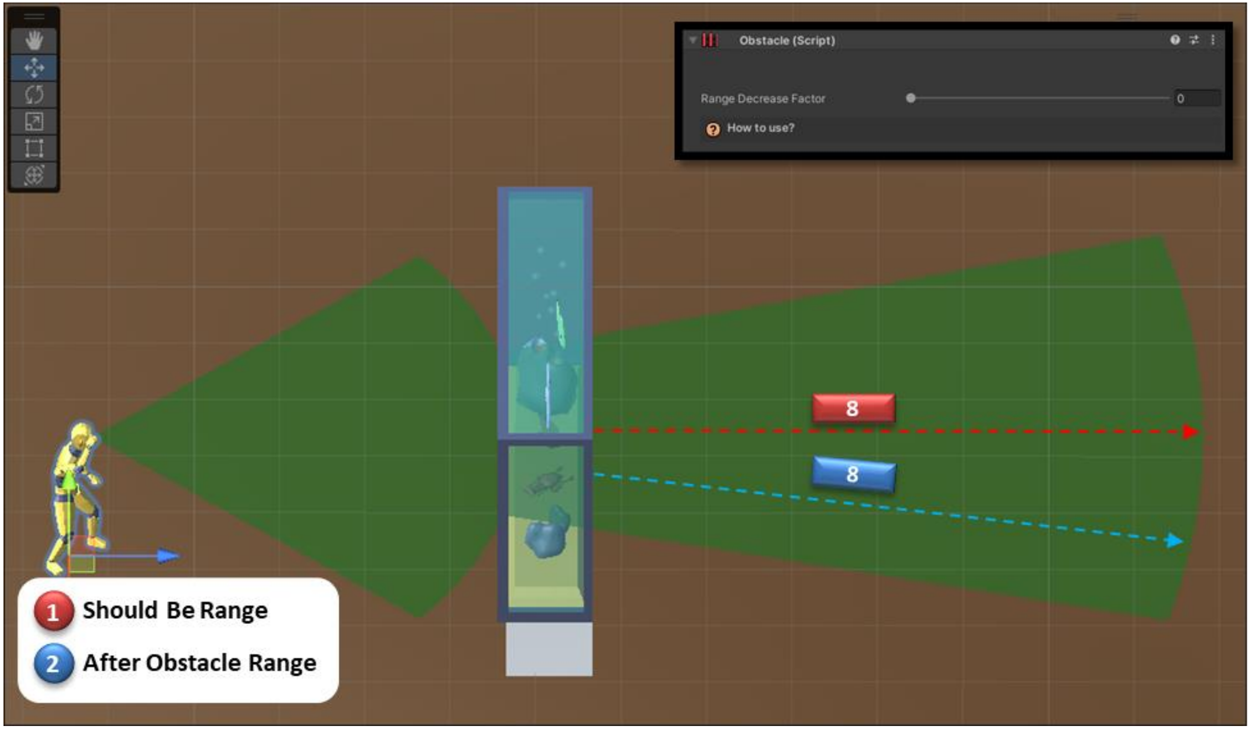
Public Methods

Set_RangeDecreaseFactor	Sets float value of Range_DecreaseFactor

Range Decrease Factor

The Range Decrease Factor is a float value that influences the vision range of a character when looking through a GameObject with an Obstacle component attached. This factor is used in the equation: New Detection Range = Original Detection Range * (1 - Range Decrease Factor). Below are examples illustrating the main principle of this mechanic:

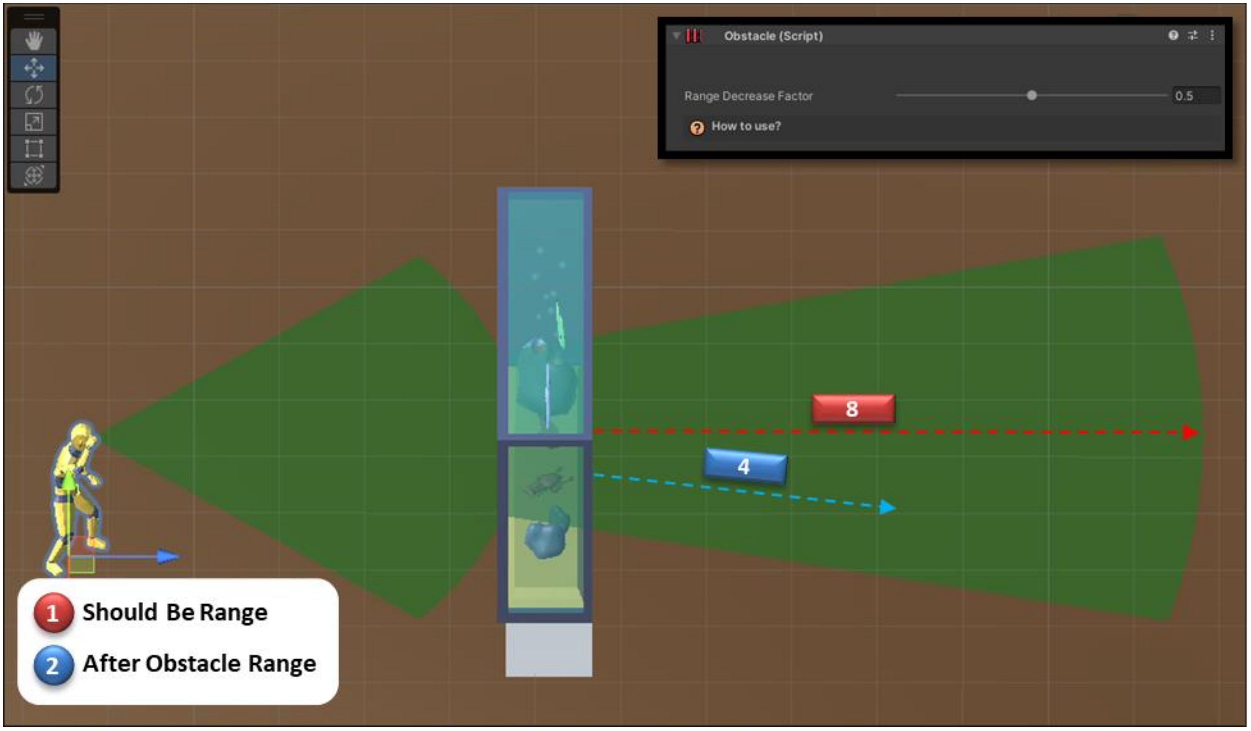
If the Range Decrease Factor equals 0, the detection range will not be modified.



Obstacle (Script)

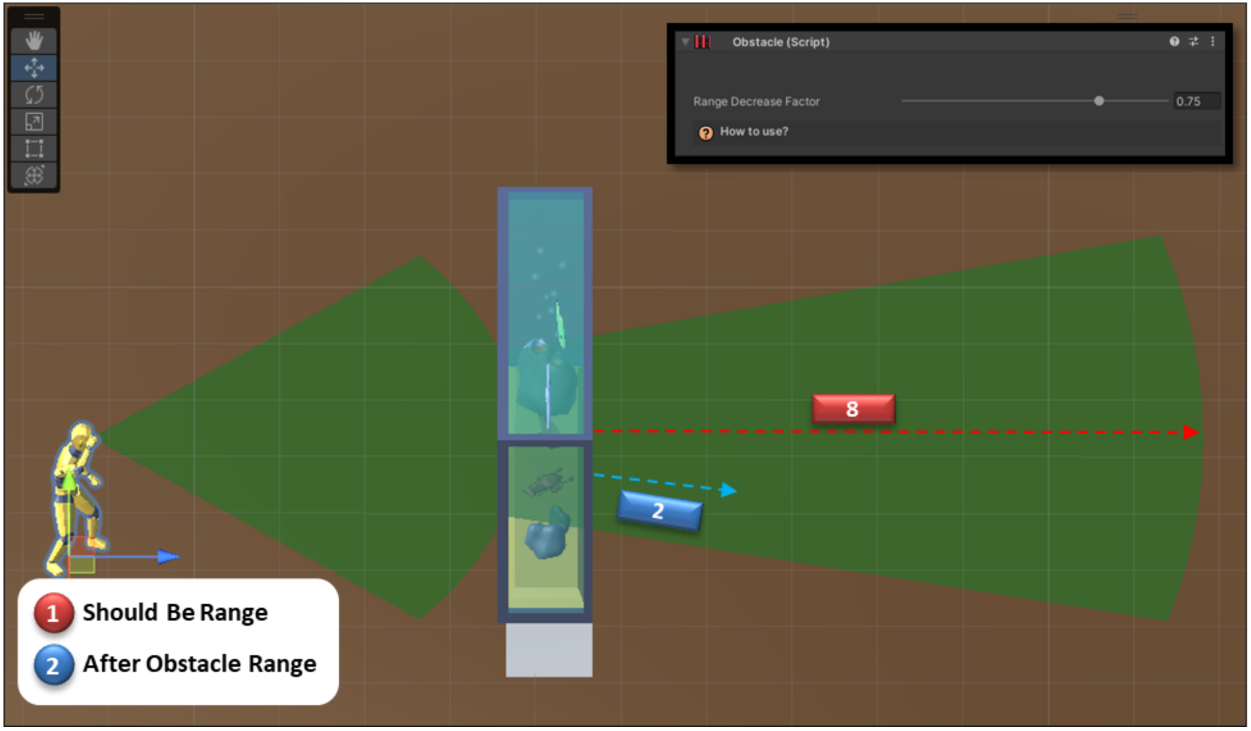
Range Decrease Factor 0

If the Range Decrease Factor equals 0.5, the remaining detection range will be halved.


Obstacle (Script)

Range Decrease Factor 0.5

Finally, if the Range Decrease Factor equals 0.75, the remaining detection range will be reduced to 25% of its original value.


Obstacle (Script)

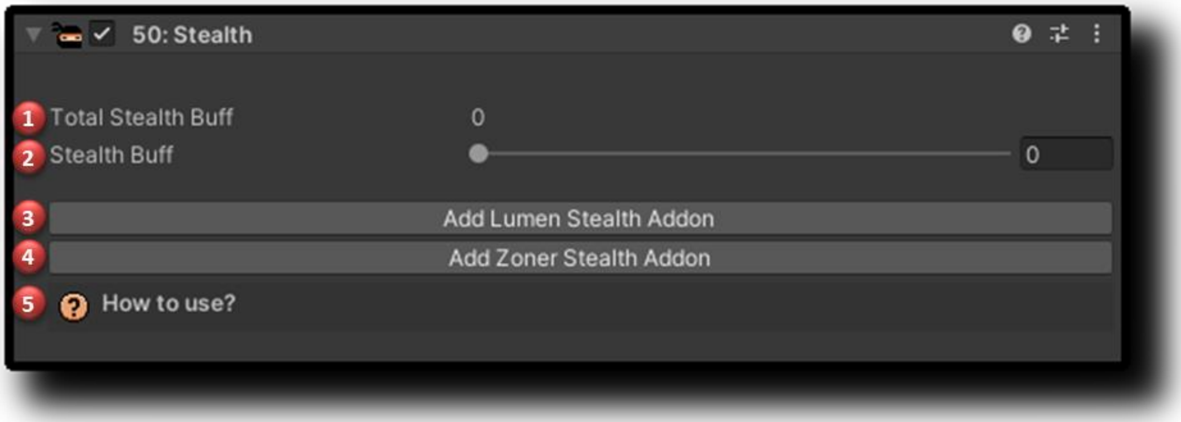
Range Decrease Factor 0.75

Stealth Component

Description

Enables the reduction of the delta See Awareness for GameObjects equipped with TargetSenses, based on the value of the Stealth Buff.

Requires **TargetSenses** Component attached.



50: Stealth

Total Stealth Buff 0

Stealth Buff 0

1	Total Stealth Buff	Current Value of Total Stealth Buff, same as value returned from Provide_TotalStealthBuff();
2	Stealth Buff	Set float value of Stealth_Buff.
3	Add Lumen Stealth Addon Button	Allows enabling Lumen Stealth Addon.
4	Add Zoner Stealth Addon Button	Allows enabling Zoner Stealth Addon.
5	‘How to use?’ Button	Displays Help Information.

Properties

Stealth_Buff	Get float value of Stealth_Buff.
Position_LumenMesurment	Get Vector3 value of Position_LumenMesurmen.
Use_LumenStealthAddon	Returns true if Use_LumenStealthAddon.
Lumen_StealthAddon	Get float value of Lumen_StealthAddon.
Use_Zoner_StealthAddon	Returns true if Use_Zoner_StealthAddon.
Zoner_StealthAddon	Get float value of Zoner_StealthAddon.
Use_CustomDetectionRadius	Returns true if Use_CustomDetectionRadius.
Radius_ZonerSteathAddon	Get float value of Radius_ZonerSteathAddon.
Use_DebugDrawRadius	Returns true if Use_DebugDrawRadius.
Limit_CollidersDetectionZoner	Get int value of Limit_CollidersDetectionZoner.
Use_ZoonerGodMode	Returns true if Use_ZoonerGodMode.
Use_LumenGodMode	Returns true if Use_LumenGodMode.

Public Methods

Set_StealthBuff	Set float value of Stealth_Buff.
Set_OffsetAxisY	Set float value of Set_OffsetAxisY.
public float Provide_TotalStealthBuff ()	Returns float value of Stealth_Buff modified with enabled Add-on Values, returns 0 if Stealth component is not enabled;
Set_LumenStealtAddon	Set status of Use_LumenStealthAddon
Set_UseZonerStealthAddon	Set status of Use_Zoner_StealthAddon
Provide_ZonerStealthAddon	Get float value of Zooner_StealthAddon.
Provide_LumenStealthAddon	Get float value of Lumen_StealthAddon.
Set_ZoonerGodMode	Set status of Use_ZoonerGodMode. When enabled Provide_TotalStealthBuff() will return 1f if Zooner_StealthAddon equals 1f.
Set_LumenGodMode	Set status of Use_LumenGodMode. When enabled Provide_TotalStealthBuff() will return 1f if Lumen_StealthAddon equals 1f.
Set_UseCustomDetectionRadius	Set status of Use_CustomDetectionRadius, when enabled ZonerStealthAddon will use Radius_ZonerSteathAddon during looking for SteathZones;
Set_RadiusZonerSteathAddon	Sets float value of Radius_ZonerSteathAddon, value will be used only when Use_CustomDetectionRadius is enabled;
Set_Limit_CollidersDetectionZoner	Sets int value of Limit_CollidersDetectionZoner. Which determines maximum amount of colliders could be detected during OverlapSphereNonAlloc;

public float Provide_TotalStealthBuff ()

This method calculates and returns the total stealth buff as a float value, considering multiple factors:

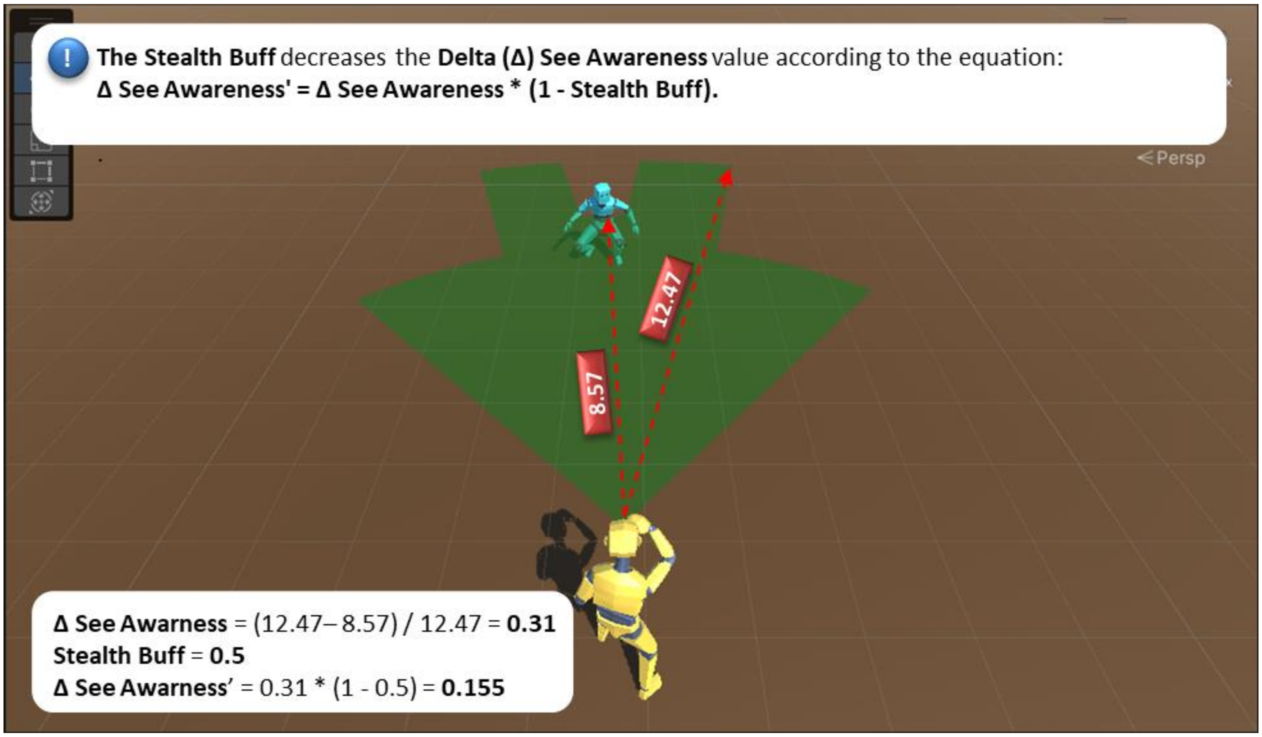
- If Use_Zoner_StealthAddon and Use_ZoonerGodMode are both enabled and Zoner_StealthAddon equals 1, the function returns 1f.
- If Use_LumenStealthAddon and Use_LumenGodMode are both enabled and Lumen_StealthAddon equals 1, the function returns 1f.

Otherwise, the method computes a weighted average of:

- Lumen_StealthAddon (if Use_LumenStealthAddon is enabled)
- Zoner_StealthAddon (if Use_Zoner_StealthAddon is enabled)
- Stealth_Buff (only if stealth is enabled)

The final value is calculated by dividing the weighted sum by the total weight.

Reduction of the delta See Awareness



The Stealth Buff decreases the **Delta (Δ) See Awareness** value according to the equation:
Δ See Awareness' = Δ See Awareness * (1 - Stealth Buff).

Δ See Awarness = (12.47– 8.57) / 12.47 = **0.31**
Stealth Buff = **0.5**
Δ See Awarness’ = 0.31 * (1 - 0.5) = **0.155**

Below few more examples of how Stealth Buff will affect **Delta See Awareness** value.

Stealth Buff = **0** → **Δ See Awarness’** = 0.31 * (1 - 0) = **0.31**
Stealth Buff = **0.5** → **Δ See Awarness’** = 0.31 * (1 - 0.5) = **0.155**
Stealth Buff = **1** → **Δ See Awarness’** = 0.31 * (1 - 1) = **0**

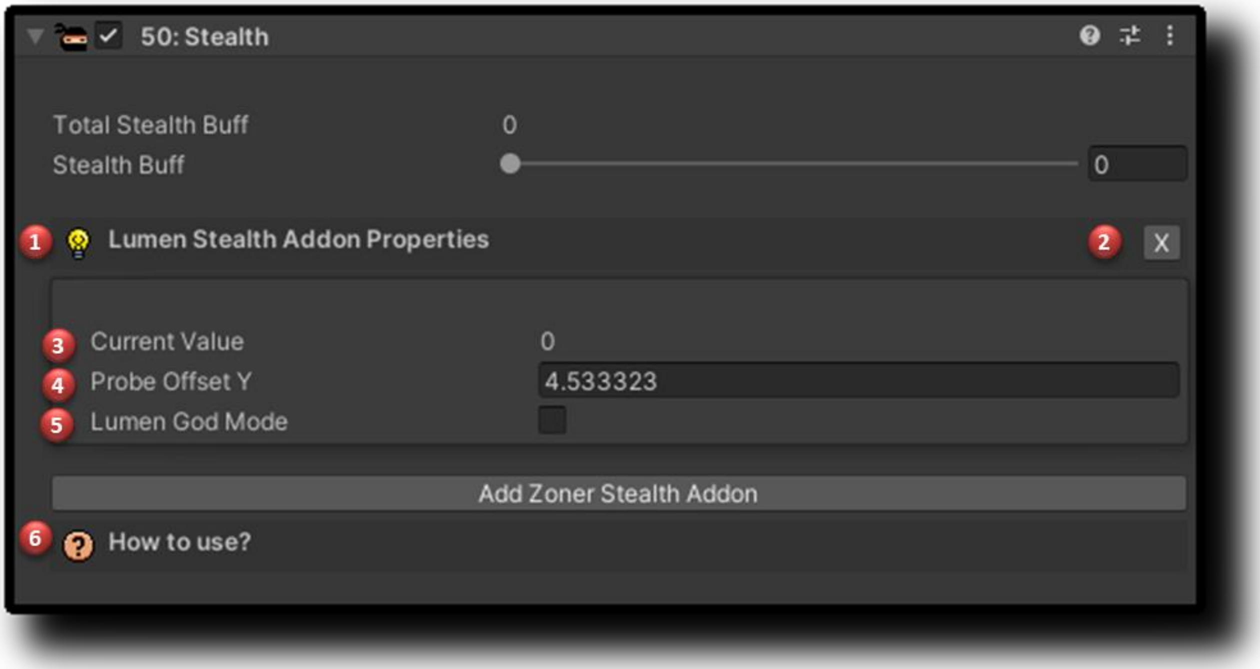
Lumen - Stealth Component Add-on

Description

Changes value of Lumen_StealthAddon according to value calculated from amount of light affecting.

GameObjects need to be equipped with **TargetSenses**.

Lumen Stealth Addon working **only during Play Mode**.



50: Stealth

Total Stealth Buff 0

Stealth Buff 0

Lumen Stealth Addon Properties

Current Value 0

Probe Offset Y 4.533323

Lumen God Mode □

1	Lumen Stealth Addon Properties Foldout	Allows showing / hiding of Lumen Stealth Addon properties.
2	Remove Button	Allows disabling Lumen Stealth Addon.
3	Lumen Stealth Addon	Displays current Lumen Stealth Addon Value (*only during Play Mode)
4	Probe Offset Y	This value, will setup high offset of light detection probe. If left for 0, during enetering play mode will, be setup for Mesh Top bound increased by .2f.
5	Lumen God Mode	When enabled, max lumen value will effect in maximum Stealth Buff Value.
6	‘How to use?’ Button	Displays Help Information.

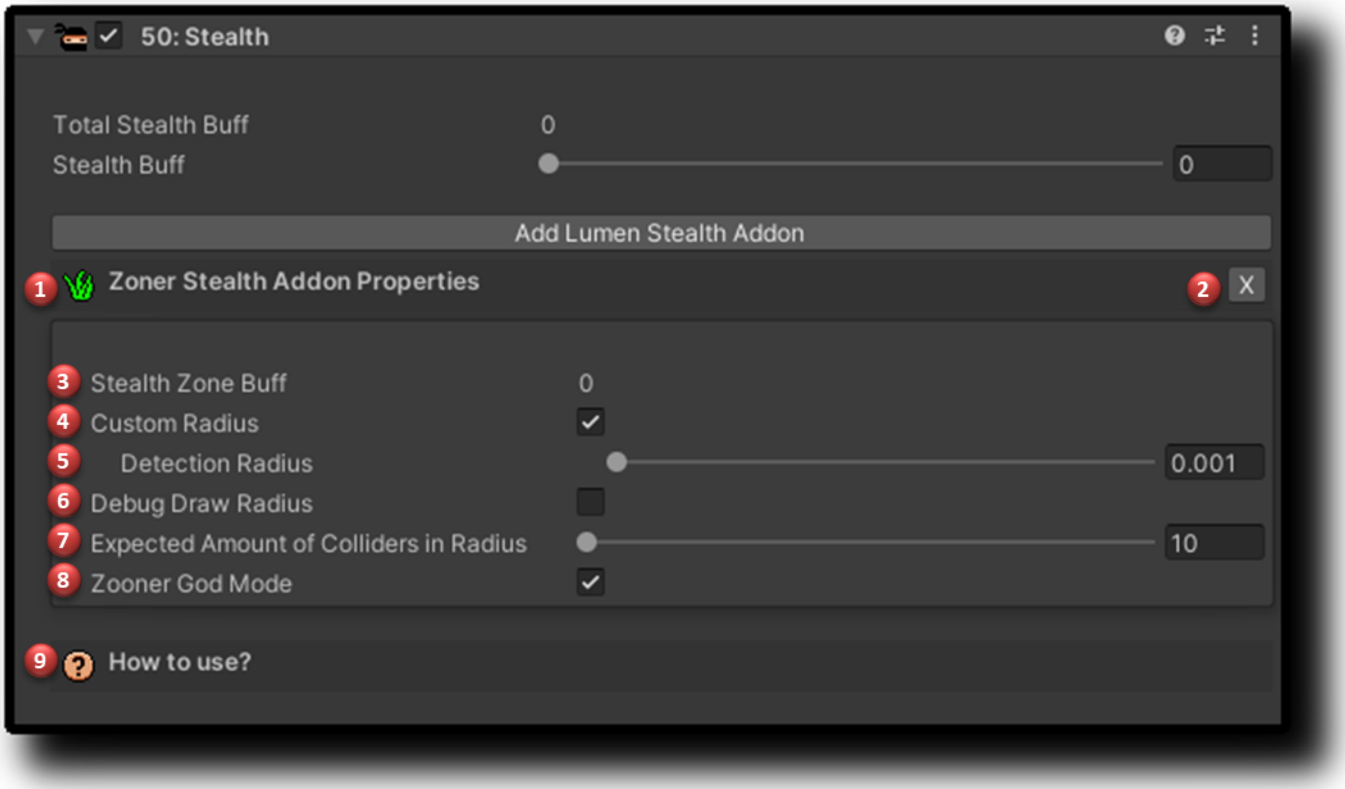
When Lumen Stealth Addon is enabled Provide_TotalStealthBuff() value will be changed while in play mode.

Zoner - Stealth Component Add-on

Description

Changes value of Zoner_StealthAddon according to value calculated as sum of StealthZones SteathBuff located in radius of character.

While **Custom Radius** enabled, checked radius will be chosen by user. While **Custom Radius** disabled radius will be calculated from attached collider / colliders bounds.



50: Stealth

Total Stealth Buff 0

Stealth Buff 0

Add Lumen Stealth Addon

1 Zoner Stealth Addon Properties 2 X

3 Stealth Zone Buff 0

4 Custom Radius ✓

5 Detection Radius 0.001

6 Debug Draw Radius □

7 Expected Amount of Colliders in Radius 10

8 Zooner God Mode ✓

9 How to use?

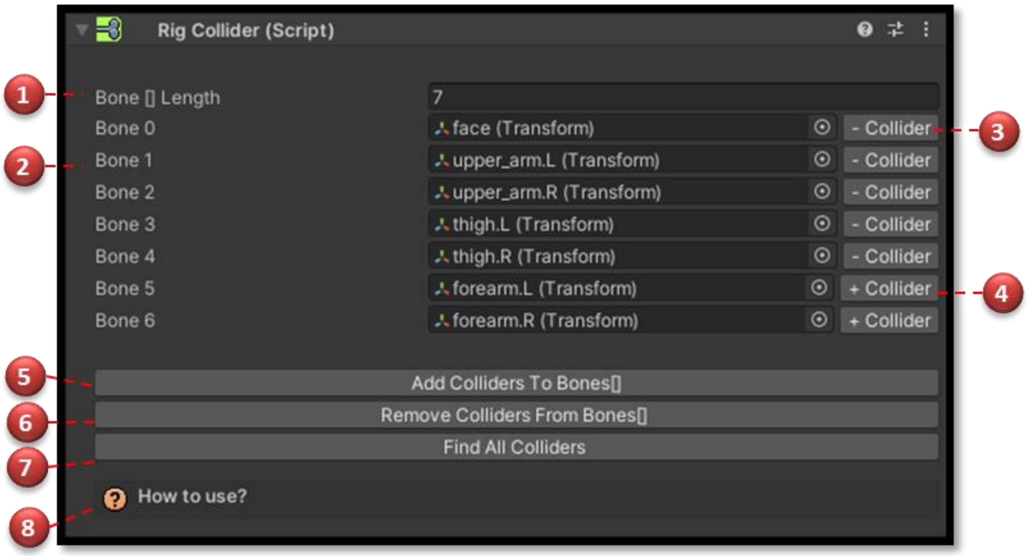
1	Zoner Stealth Addon Properties Foldout	Allows showing / hiding of Zoner Stealth Addon properties.
2	Remove Button	Allows disabling Zoner Stealth Addon.
3	Zoner Stealth Buff	Displays current Zoner Stealth Addon Value (*only during Play Mode)
4	Custom Radius	When enabled, allows determine Radius_ZonerSteathAddon of checking area around for colliders with StealthZones. When disabled Radius_ZonerSteathAddon will be calculated from

		attached collider / colliders bounds.
5	Detection Radius	Allows setting up float value of radius for checking area around for colliders with SteathZones.
6	Debug Draw Radius	When enabled, with gizmos turn on, allows for graphic presentation of Radius_ZonerSteathAddon.
7	Expected Amount of Colliders in Radius	Allows setting up int value of Limit_CollidersDetectionZoner. Which determines maximum amount of colliders could be detected during OverlapSphereNonAlloc.
8	Zooner God Mode	When enabled, max zooner value will effect in maximum Stealth Buff Value.
9	‘How to use?’ Button	Displays Help Information.

RigCollider Component

Description

This component enables the management of colliders attached to the character rig selected by the end user. Allows for preparing Characters for more detailed detection.



Rig Collider (Script)

Bone [] Length 7

Bone 0 face (Transform) - Collider

Bone 1 upper_arm.L (Transform) - Collider

Bone 2 upper_arm.R (Transform) - Collider

Bone 3 thigh.L (Transform) - Collider

Bone 4 thigh.R (Transform) - Collider

Bone 5 forearm.L (Transform) + Collider

Bone 6 forearm.R (Transform) + Collider

1	Bone Array Length	Allows setting up length of Bone Array
2	Bone Array Elements	Allows assigning of Transform chosen as future Colliders
3	“- Collider”	Remove existing Collider
4	“+ Collider”	Generate Collider
5	Add Colliders To Bones[] Button	Will Generate all missing Colliders to Bone Array
6	Remove Colliders From	Will Destroys all Colliders from Bone Array

	Bones[] Button	
7	Find all Colliders	Will find all Colliders
8	‘How to use?’ Button	Displays Help Information.

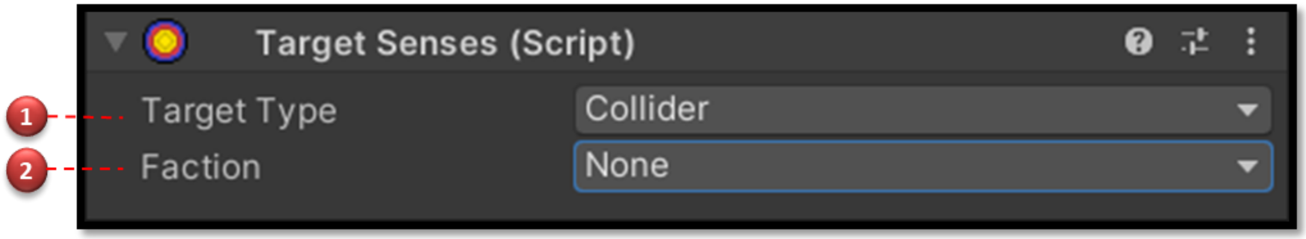
TargetSenses Component

Description

This component is necessary for characters with the Senses component attached to perceive it as an entity that needs to be tracked by awareness value.

Requires **either** a **Collider** or a **RigCollider** component to be attached.

Allows usage of faction system (introduced in 1.02).



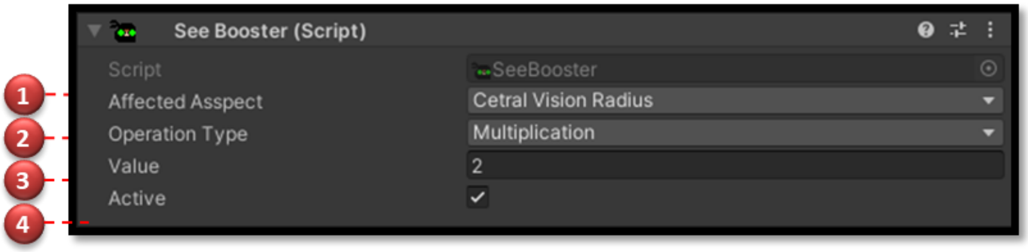
Target Type: Collider
Faction: None

1	Target Type	Allows setting up target type. Chose 'Collider' (performance weiss solution) or 'RigCollider’.
2	Faction	Allows choosing faction. If not needed, leave as 'None'. If need to add more Factions or change names edit FactionNames.cs file.

See Booster Component

Description

This component allows for modification of See Sensor enable in Senses Component. Could modify delta See Awareness, Central Vision Range or Peripheral Vision Range. Supported types of operation are Addition, Subtraction, Multiplication, and Division.



See Booster (Script)

Script: SeeBooster

Affected Asspect: Cetral Vision Radius

Operation Type: Multiplication

Value: 2

Active: ✓

1	Affected Aspect	Sets enum of Affected Asspect (delta See Awareness, Central Vision Range or Peripheral Vision Range).
2	Operation Type	Sets enum of Operation Type (Addition, Subtraction, Multiplication, and Division).
3	Value	Sets float value of SeeBooster.
4	Active	Sets bool state of “Active”.

Properties

AffectedAspect	Get enum of Affected Asspect.
OperationType	Get enum of Operation Type.
Value	Get float value of SeeBooster.
Active	Get bool state of “Active”.

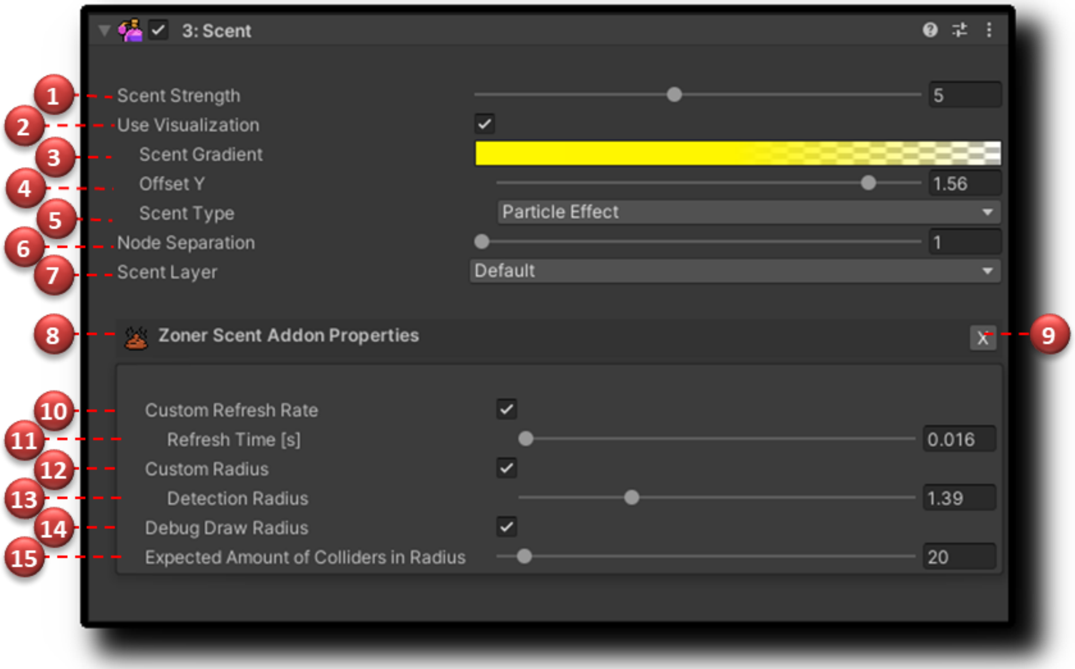
Public Methods

Set_AffectedAsspect	Set enum of Affected Asspect.
Set_OperationType	Set enum of Operation Type.
Set_Value	Set float value of SeeBooster.
Set_ActiveState	Set bool state of “Active”.

Scent Component

Description

This component is necessary for characters with the **TargetSenses** component attached to spread trail of scent // smell behind him. Scent will be released only during movement. This component when “**Use Visualization**” enabled will create Scent Visualization during Start ().


3: Scent

Scent Strength 5

Use Visualization ✓

Scent Gradient

Offset Y 1.56

Scent Type Particle Effect

Node Separation 1

Scent Layer Default

Zoner Scent Addon Properties X

Custom Refresh Rate ✓

Refresh Time [s] 0.016

Custom Radius ✓

Detection Radius 1.39

Debug Draw Radius ✓

Expected Amount of Colliders in Radius 20

1	Scent Strength	Sets float value of Scent Strength.
2	Use Visualization	Click to enable/disable scent visualization.
3	Scent Gradient	Sets gradient value of scent visualization.
4	Offset Y	Sets position offset of scent visualization.
5	Node Separation	Sets amount of distance between released Scents Nodes. Lower value will result in denser trail with bigger amount of Scent Nodes.
6	Scent Layer	Sets layer for scent.
7	Scent Type	Chose TrailRender or ParticleSystem based scent visualization.
8	Zoner Scent Addon Properties	Click to show/hide ZonerScentAddon properties.
9	X	Click to disable Zoner Scent Addon.
10	Custom Refresh Rate	When enabled, allows determine refresh rate of checking area around for colliders with ScentZoneDistractor. When disabled refresh will be conducted during each Update iteration.
11	Refresh Time [s]	Allows setting up float value of custom refresh time in seconds.
12	Custom Radius	When enabled, allows determine radius of checking

		area around for colliders with ScentZoneDistractor. When disabled radius will be calculated from attached collider / colliders bounds.
13	Detection Radius	Allows setting up float value of radius for checking area around for colliders with ScentZoneDistractor.
14	Debug Draw Radius	When enabled, with Gizmos Turn On, allows for graphic presentation of radius for checking area around for colliders with ScentZoneDistractor.
15	Expected Amount of Colliders in Radius	Small Value could be used when is not many colliders in radius area, High Value with multiple colliders in radius area.

Properties

ScentStrength	Gets float value of ScentStrength.
UseVisualisation	Gets bool value of UseVisualization.
ScentColorGradient	Gets Gradient value of ScentColorGradient.
NodeSeparation	Gets float value of NodeSeparation.
ScentLayer	Gets LayerMask value of ScentLayer.
Use_Zoner_ScentAddon	Gets bool value of Use_Zoner_ScentAddon.
Use_CustomRefreshRate	Gets bool value of Use_CustomRefreshRate.
CooldownRefreshRate	Gets float value of CooldownRefreshRate.
Use_CustomDetectionRadius	Gets bool value of Use_CustomDetectionRadius.
Radius_ZonerScentAddon	Gets float value of Radius_ZonerScentAddon.
Use_DebugDrawRadius	Gets bool value of Use_DebugDrawRadius.
VisualisationOffsetY	Gets float value of VisualisationOffsetY.
Limit_CollidersDetectionZoner	Gets int value of Limit_CollidersDetectionZoner.

Public Methods

Set_ScentStrength	Sets float value of ScentStrength.
Set_UseVisualisation	Sets bool value of UseVisualization.
Set_ScentColorGradient	Sets Gradient value of ScentColorGradient.
Set_NodeSeparation	Sets float value of NodeSeparation.
Set_UseZonerScentAddon	Sets bool value of Use_Zoner_ScentAddon.
Set_UseCustomRefreshRate	Sets bool value of Use_CustomRefreshRate.
Set_CooldownRefreshRate	Sets float value of CooldownRefreshRate.
Set_UseCustomDetectionRadius	Sets bool value of Use_CustomDetectionRadius.
Set_Radius_ZonerScentAddon	Sets float value of Radius_ZonerScentAddon.
Set_UseDebugDrawRadius	Sets bool value of Use_DebugDrawRadius.
Set_Limit_CollidersDetectionZoner	Sets int value of Limit_CollidersDetectionZoner.
Set_ScentLayer	Sets LayerMask value of ScentLayer.
Set_VisualisationOffsetY	Sets float value of VisualisationOffsetY.

Zoner - Scent Component Add-on

Description

This Add-On is part of the Scent Component. When enabled, it will search for GameObjects with attached Scent Zone Distractor Components. If any of the results contain a Scent Zone Distractor with the Scent Modification set to “BlockScentEmission,” scent emission will be blocked while within range of the mentioned GameObject. If any of the results contain a Scent Zone Distractor with the Scent Modification set to “ChangeScentStrength,” the effects will be summed among all such results, and the final modified scent strength will be applied during scent release.

Scent Zone Distractor Component

Description

The Scent Zone Distractor Component is a specialized component designed to influence and modify the behavior of scent emissions within a defined area or zone. When attached to a GameObject, this component can either block or alter the strength of scents emitted by nearby objects, based on its configuration.

Key Methods

```
public bool DidSee(
                                        TargetSenses _targetSenses,
                                        int _requiredAwarness,
                                        out float _seeAwarness,
                                        out float _distanceSQR,
                                        out float _angle)
```

Description

Returns true if requirements were match (so amount off see awareness acquired for _targetSense was high enough).
Provides _seeAwarness, _distance, _angle as out parameter.

In (search requirements)

1	_targetSenses	Allows for choosing TargetSense to check.
2	_requiredAwareness	Allows for choosing minimum required Awareness.

Out

1	_seeAwarness	Returns float value of current seeAwarness.
2	_distanceSQR	Returns float value of sqrt Distance to TargetSenses.
3	_angle	Returns float value of Angle towards TargetSenses.

```
public bool DidHear (
                                        TargetSenses _targetSenses,
                                        int _requiredAwarness,
                                        out float _hearAwarness,
                                        out Vector3 _soundLastPosition)
```

Description

Returns true if requirements were match (so amount off hear awareness acquired for _targetSense was high enough).
Provides _hearAwarness, _ soundLastPosition as out parameter.

In (search requirements)

1	_targetSenses	Allows for choosing TargetSense to check.
2	_requiredAwareness	Allows for choosing minimum required Awareness.

Out

1	_hearAwarness	Returns float value of current hearAwarness.
2	_soundLastPosition	Returns Vector3 to last know released noise position.

```
public bool DidSmell(
                                        TargetSenses _targetSenses,
                                        int _requiredAwarness,
                                        out float _smellAwarness,
                                        out Vector3 _scentLastPosition
                                        out float _scentStrength)
```

Description

Returns true if requirements were match (so amount off smell awareness acquired for _targetSense was high enough).
Provides _smellAwarness, _scentLastPosiotn as out parameter.

In (search requirements)

1	_targetSenses	Allows for choosing TargetSense to check.
2	_requiredAwareness	Allows for choosing minimum required Awareness.

Out

1	_smellAwarness	Returns float value of current seeAwarness.
2	_scentLastPosition	Vector3 to last know scent position.
3	_scentStrength	Returns float value of last know scent strength.

```
public bool WasAnythingSeen(
                            float _requiredAwarness,
                            out TargetSenses _targetSenses,
                            out float _currentAwarness,
                            out float _distanceSQR,
                            out float _angle,
                            FactionNames[] _factionNames
                            SeeSenseSearchType _searchType
                            bool _needToBeInCentralRange = true)
```

Description

Returns true if requirements were match (so amount off see awareness acquired for some _targetSense was high enough).
Provides _targetSenses, _currentAwarness, _distanceSQR, _angle as out parameter.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Array allows choosing which factions will be checked for results.
3	_searchType	Allows choosing SeeSenseSearchType.
4	_needToBeInCentralRange	Allows deciding did TargetSense need to be in central view range.

Out

1	_targetSenses	Returns TargetSenses of “Seen” object.
2	_currentAwarness	Returns float value of current Awarness.
3	_distanceSQR	Returns float value of sqrt Distance to TargetSenses.
4	_angle	Returns float value of Angle towards TargetSenses.

```
public bool WasAnythingHear(
                                        float _requiredAwarness,
                                        out TargetSenses _targetSenses,
                                        out float _currentAwarness,
                                        out Vector3 _soundLastPostion,
                                        FactionNames[] _factionNames
                                        HearSenseSearchType _hearSearchType)
```

Description

Returns true if requirements were match (so amount off awareness acquired for some _targetSense was high enough).
Provides _targetSenses, _currentAwarness, _soundLastPostion as out parameter.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Array allows choosing which factions will be checked for results.
3	_hearSearchType	Allows for choosing HearSerachType

Out

1	_targetSenses	Returns TargetSenses of “Hear” object.
2	_currentAwarness	Returns float value of current Awarness.
3	_soundLastPostion	Returns Vector3 to last know released noise position.

```
public bool WasAnythingSmell(
                                        float _requiredAwarness,
                                        out TargetSenses _targetSenses,
                                        out float _currentAwarness,
                                        out Vector3 _scentLastPosition,
                                        out float _scentStrength,
                                        FactionNames[] _factionNames
                                        SmellSenseSearchType _hearSearchType)
```

Description

Returns true if requirements were match (so amount off awareness acquired for some _targetSense was high enough).
Provides _targetSenses, _currentAwarness, _scentLastosition, _scentStrength as out parameter.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Array allows choosing which factions will be checked for results.
3	_hearSearchType	Allows for choosing HearSerachType

Out

1	_targetSenses	Returns TargetSenses of “Smell” object.
2	_currentAwarness	Returns float value of current Awarness.
3	_soundLastPostion	Returns Vector3 to last know released noise position.

```
public List<TargetSenses> GetDetectedTargetList(
                                        FactionNames[] _factionNames,
                                        float _requiredAwareness = 1f,
                                        bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses above required Awareness value.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required array of FactionNames.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

```
public List<TargetSenses> GetDetectedTargetList(
                                    FactionNames _factionNames = FactionNames.None,
                                   float _requiredAwareness = 1f,
                                    bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses above required Awareness value.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required FactionNames. By default set as FactionNames.None.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

```
public List<TargetSenses> GetDetectedTargetList(
                                    out List<float> _awareness
                                    FactionNames[] _factionNames,
                                    float _requiredAwareness = 1f,
                                    bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses above required Awareness value.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required array of FactionNames.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

Out

1	_awareness	Returns List<float> value of Awareness of TargetSenses meting search requirements;

```
public List<TargetSenses> GetDetectedTargetList(
                                   out List<float> _awareness
                                   FactionNames _factionNames = FactionNames.None,
                                   float _requiredAwareness = 1f,
                                   bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses above required Awareness value.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required FactionNames. By default set as FactionNames.None.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

Out

1	_awareness	Returns List<float> value of Awareness of TargetSenses meting search requirements;

```
public List<TargetSenses> GetRememberedTargetList(
                                   FactionNames[] _factionNames,
                                   float _requiredAwareness = 1f,
                                   bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness bellow required value but higher than 0.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required array of FactionNames.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

```
public List<TargetSenses> GetRememberedTargetList(
                              FactionNames _factionNames = FactionNames.None,
                              float _requiredAwareness = 1f,
                              bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness bellow required value but higher than 0.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required FactionNames. By default set as FactionNames.None.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

```
public List<TargetSenses> GetRememberedTargetList(
                              out List<float> _awareness
                              FactionNames[] _factionNames,
                              float _requiredAwareness = 1f,
                              bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness bellow required value but higher than 0.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required array of FactionNames.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

Out

1	_awareness	Returns List<float> value of Awareness of TargetSenses meting search requirements;

```
public List<TargetSenses> GetRememberedTargetList(
                                FactionNames _factionNames = FactionNames.None,
                                out List<float> _awareness
                                float _requiredAwareness = 1f,
                                bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness bellow required value but higher than 0.

In (search requirements)

1	_requiredAwareness	Allows for choosing minimum required Awareness.
2	_factionNames	Allows for choosing required FactionNames. By default set as FactionNames.None.
3	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

Out

1	_awareness	Returns List<float> value of Awareness of TargetSenses meting search requirements;

```
public List<TargetSenses> GetAllTargetList(
                                FactionNames[] _factionNames,
                                bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness above 0.

In (search requirements)

1	_factionNames	Allows for choosing required array of FactionNames.
2	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

```
public List<TargetSenses> GetAllTargetList(
                                    FactionNames _factionNames = FactionNames.None,
                                    bool _needToBeInCentralRange = true)
```

Description

Provides List of TargetSenses with Awareness above 0.

In (search requirements)

1	_factionNames	Allows for choosing required FactionNames. By default set as FactionNames.None.
2	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.