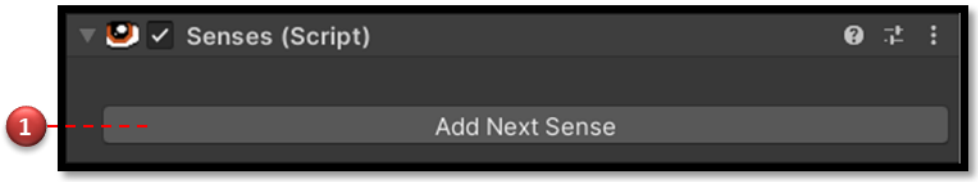
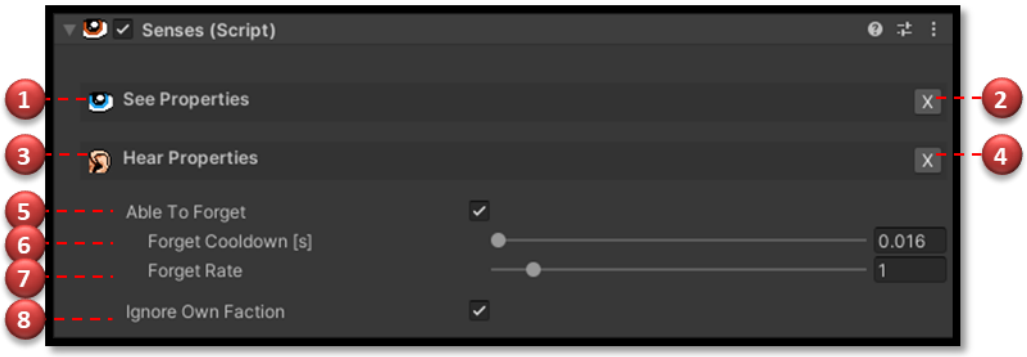
Senses Component

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.



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	Able To Forget	Able To Forget allows the setting of the Use_Forgetting state. If enabled, Awareness will decrease by the Forget Rate within the Forget Cooldown [s] time interval.
6	Forget Cooldown [s]	Allows setting up float value of time interval between each Awareness decrease.
7	Forget Rate	Allows setting up float value of Awareness decreased during each time interval.
8	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 Rate** within the **Forget Cooldown [s]** time interval, 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.

Public Properties

Use_See	Returns true if uses See Sensor.
Eyes	Get transform Eyes.
Use_Hear	Returns true if uses Hear Sensor.
IgnoreLayers	Get Ignored Layers
Use_CustomRefreshRate	Returns true if uses custom refresh rate.
Cooldown_SeeTimeAmount	Get float value of Cooldown_SeeTimeAmount
Use_Forgeting	Returns true if uses forgetting.
Cooldown_Forgeting	Get float value of Cooldown_Forgeting
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 postiion offset.
Hear_Sensitivity	Returns enum value of Hear_Sensitivity.
TagetSensesWasDetected	Returns true if search requirements were match. Provides TargetSenses as out parameter.
WasDetected	Returns true if provided TargetSenses was detected.
GetDetectedTargetList	Provides List of TargetSenses above required Awareness value.
GetRememberedTargetList	Provides List of TargetSenses below required Awareness value but above zero.
~~GetForgetedTargetList~~	Removed in 1.04
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

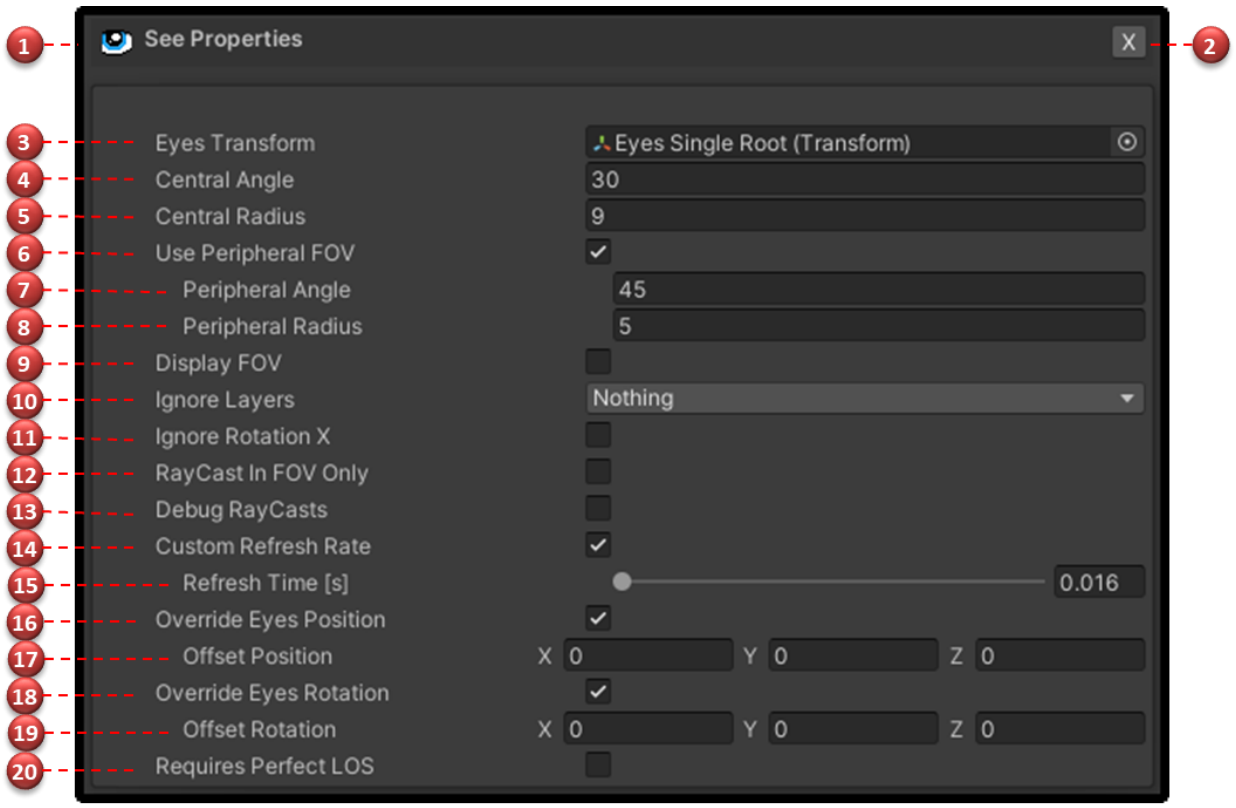
Set_Eyes	Sets transform Eyes.
Set_UseSee	Sets state of bool Use_See.
Set_UseHear	Sets state of bool Use_Hear.
Set_UseCustomRefreshRate	Sets state of bool Use_CustomRefreshRate.
Set_CooldownSeeTimeAmount	Sets float value of Cooldown_SeeTimeAmount.
Set_UseForgeting	Sets state of bool Use_Forgeting.
Set_CooldownForgeting	Sets float value of Cooldown_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

See - Senses Sensor

Description

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

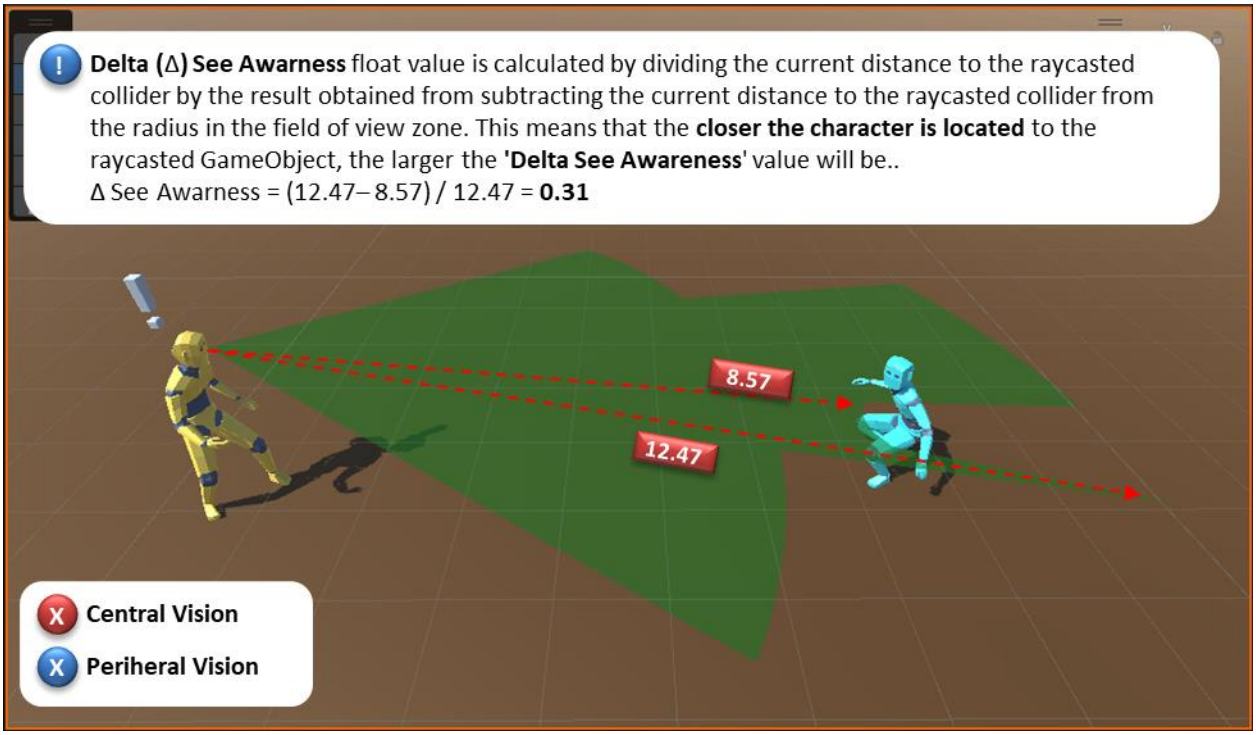
See Sensor could be enabled or disabled either through inspector tab or code.
See Sensor gather **See Awareness** of **GameObject** with attached **TargetSenses** component.
Properties of See Sensor could be setup either through inspector tab or code.

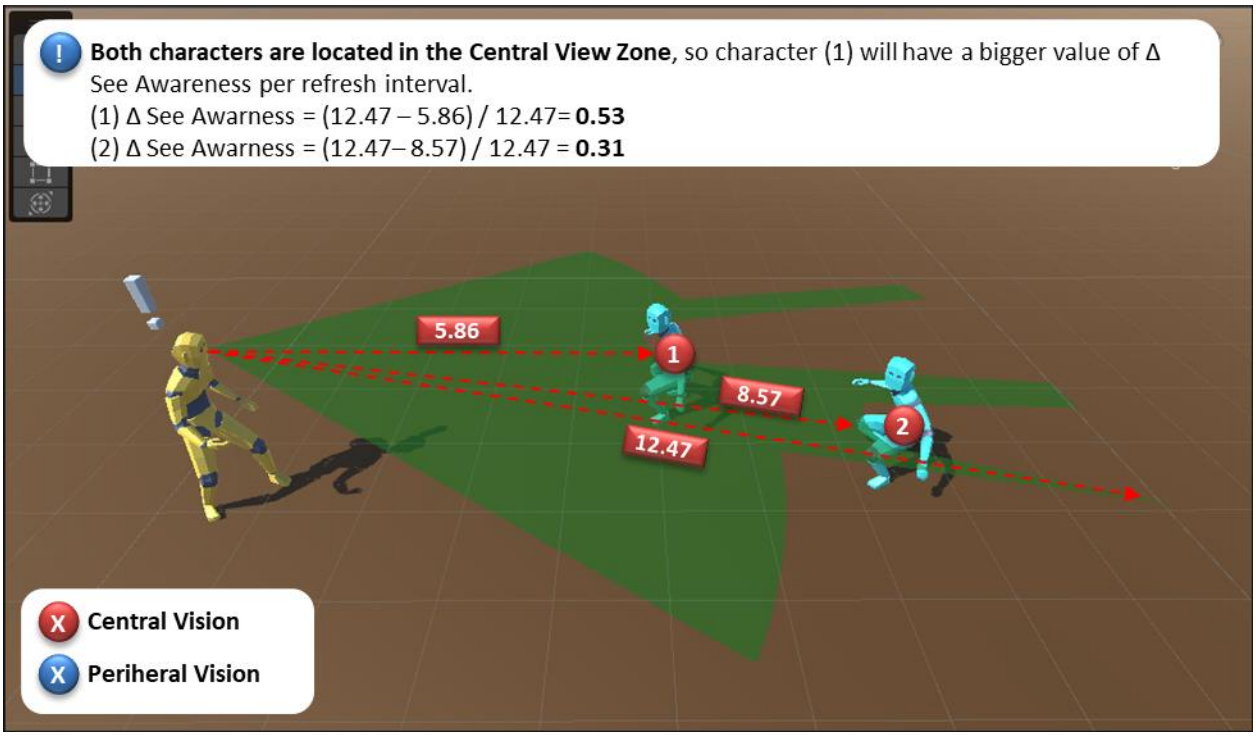


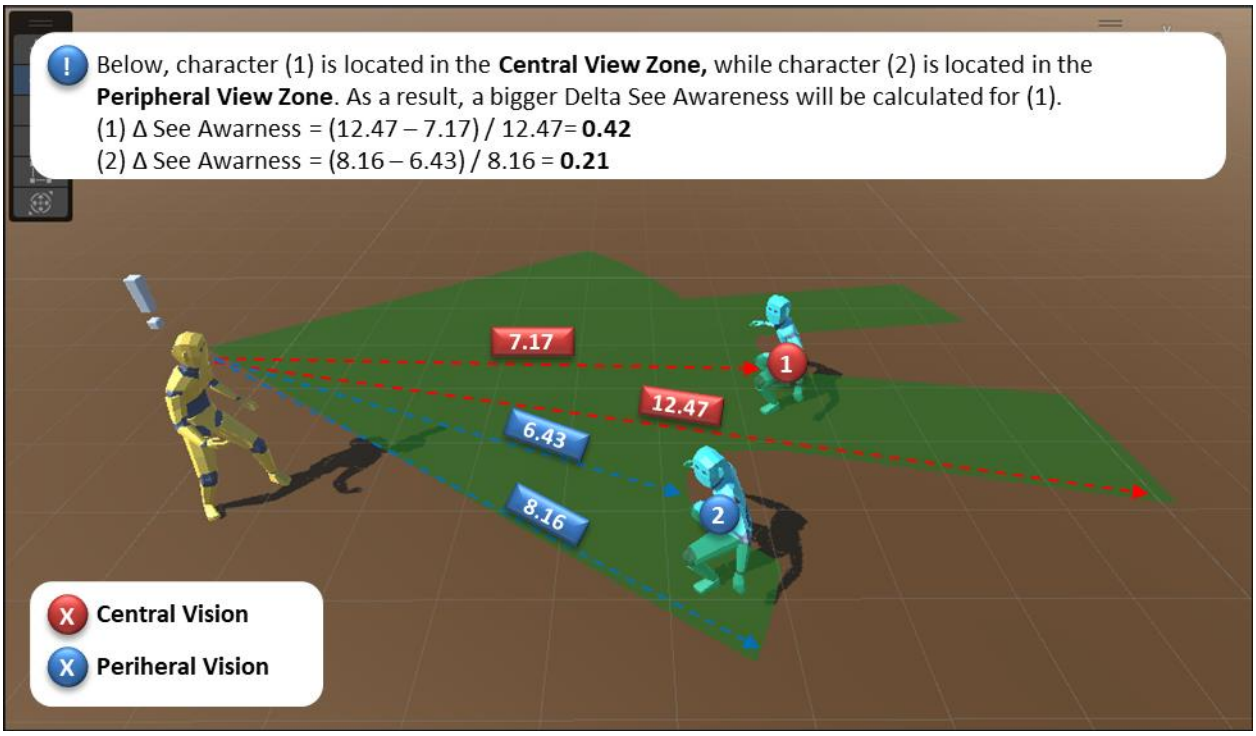
1	See Properties Foldout	Allows showing / hiding of See Sensor properties.
2	Remove Button	Allows disabling See Sensor.
3	Eyes Transform	Allows assign of Transform used as Eyes Transform. Eyes Transform position is used during radius, and angle calculations. Furthermore serves as raycast origin point 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	Custom Refresh Rate	When disabled, each sense sensor refreshes during every frame of the update. However, when activated, it permits the use of a custom refresh time, defined by the Refresh Time [s] parameter.
15	Refresh Time [s]	Allows setting up float value of senses refresh time in seconds.
16	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)
17	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.
18	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)
19	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.
20	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.

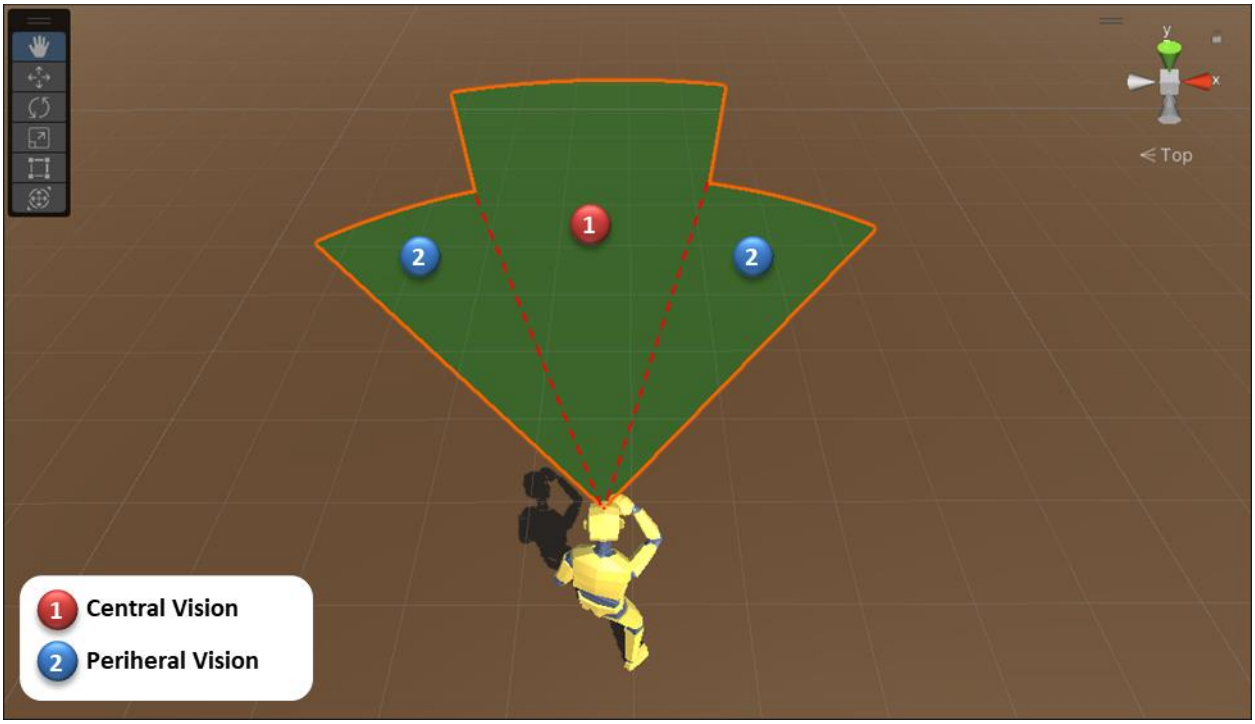
Delta See Awareness





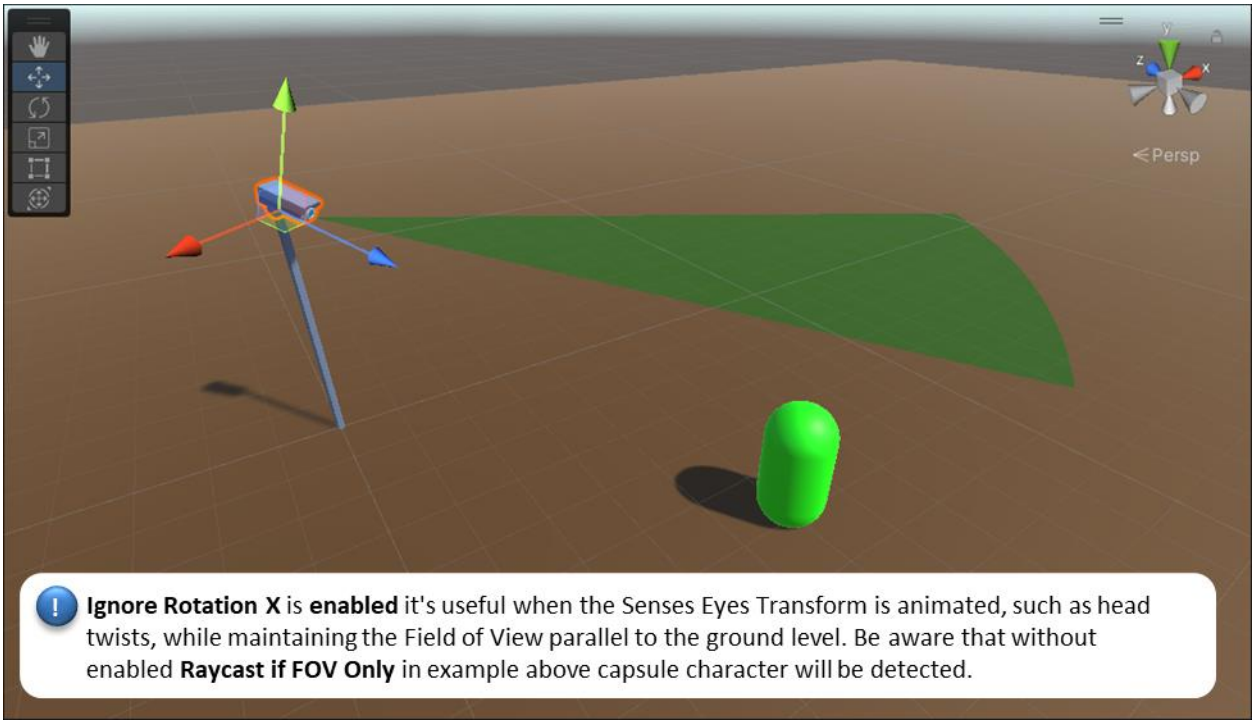


Field of View Zones

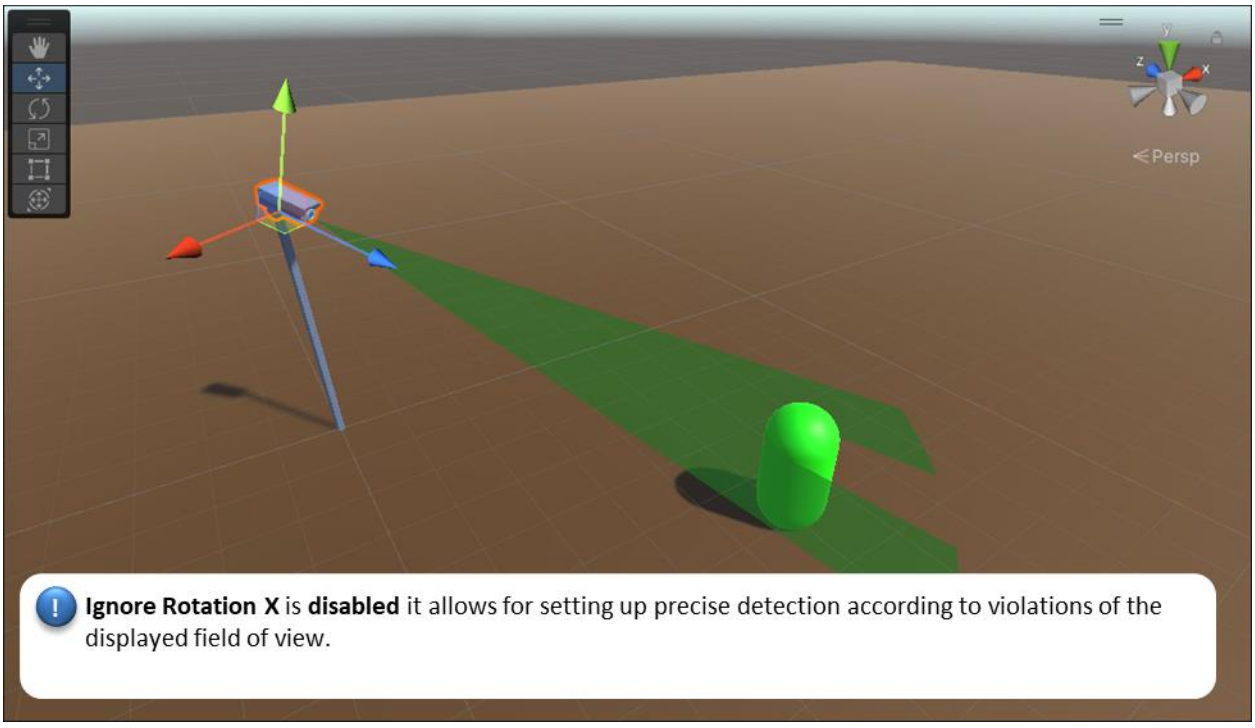


.

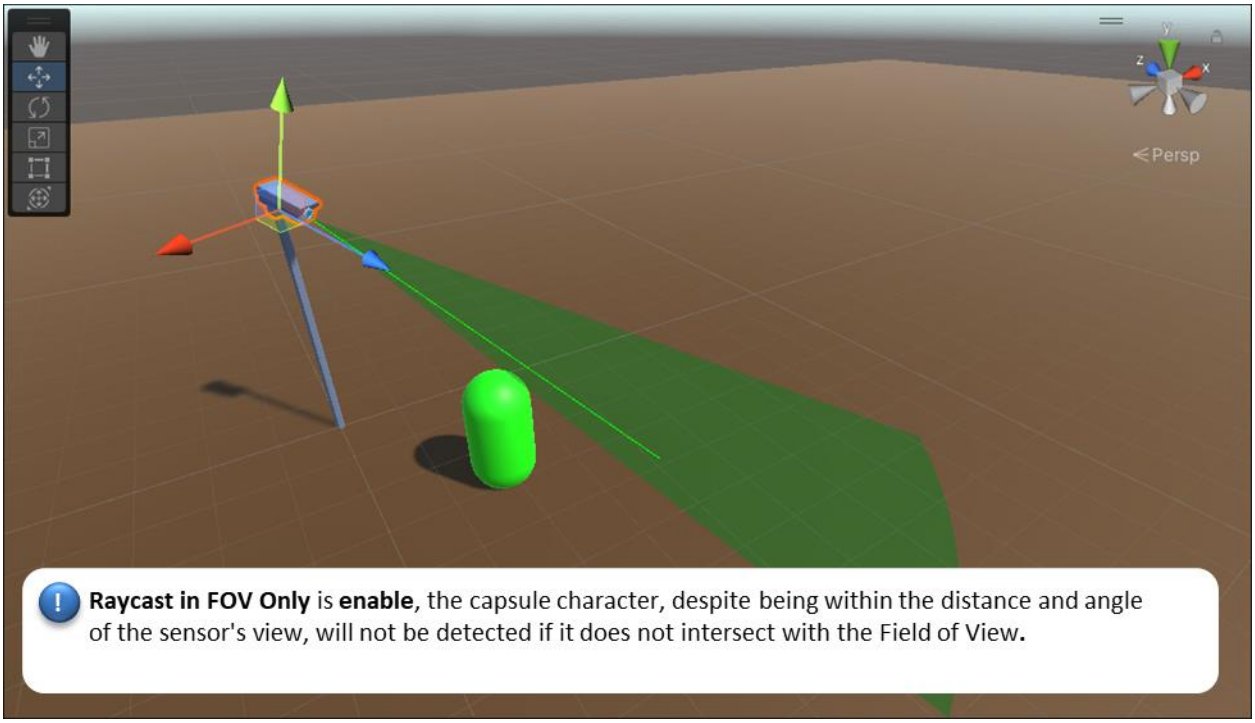
Ignore Rotation X (enabled).



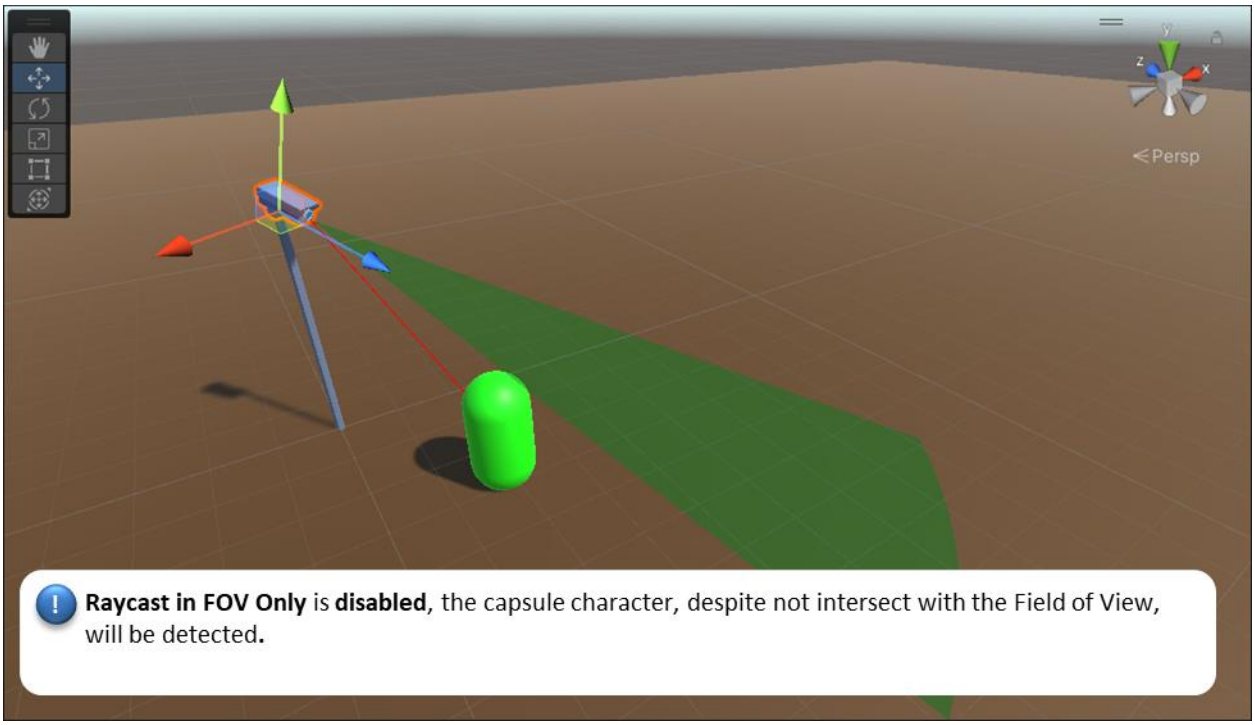
Ignore Rotation X (disabled).



Raycast in FOV Only (enabled).



Raycast in FOV Only (disabled).



Public Properties

Use_See	Returns true if uses See Sensor.
Eyes	Get transform Eyes.
IgnoreLayers	Get Ignored Layers
Use_CustomRefreshRate	Returns true if uses custom refresh rate.
Cooldown_SeeTimeAmount	Get float value of Cooldown_SeeTimeAmount
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

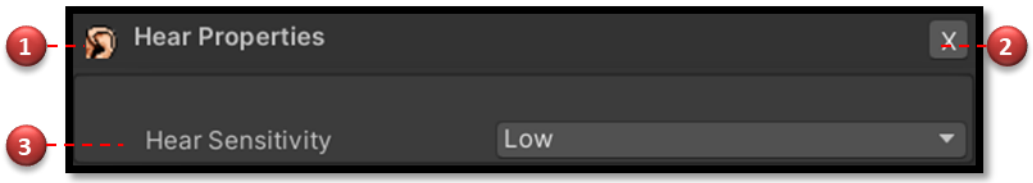
Set_UseSee	Sets state of bool Use_See.
Set_Eyes	Sets transform Eyes.
Set_UseCustomRefreshRate	Sets state of bool Use_CustomRefreshRate.
Set_CooldownSeeTimeAmount	Sets float value of Cooldown_SeeTimeAmount.
Set_CentralVisionAngle	Sets float value of Central_VisionAngle.
Set_CentralVisionRadius	Sets float 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

Noise Component

Description

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.

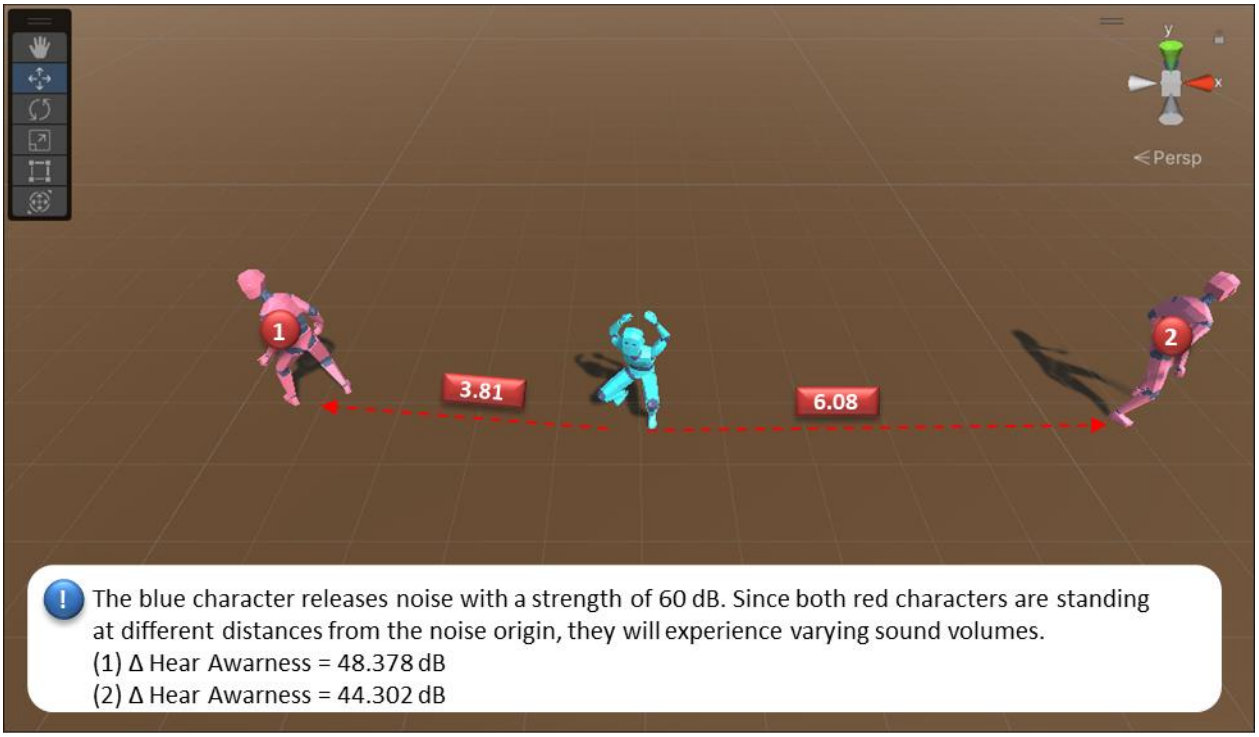
void Release_Noise (float _noiseValue)

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.

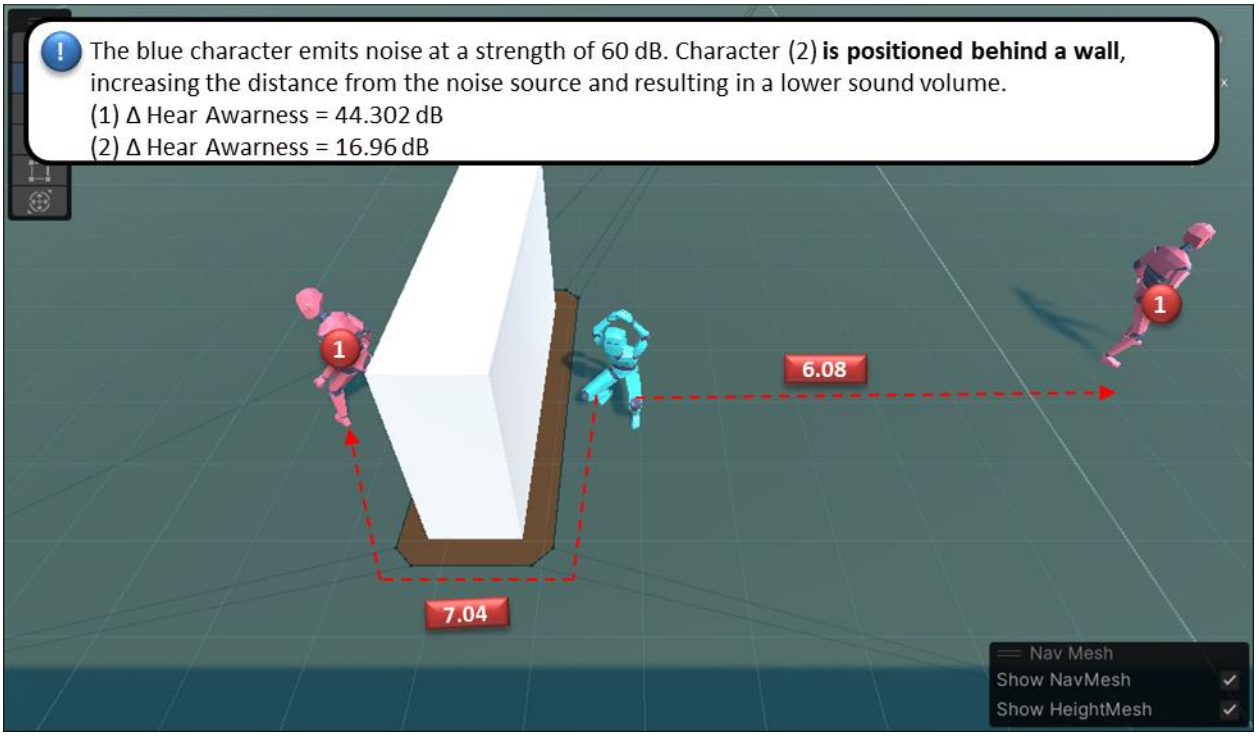
The _noiseValue parameter represents the real-world sound power in decibels (dB). Example values are provided for reference.

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



Use_NaveMesh



Obstacle Component

Description

Mitigates the detection of objects located behind it.

Properties

`Range_DecreaseFactor`	`Get float value of Range_DecreaseFactor`

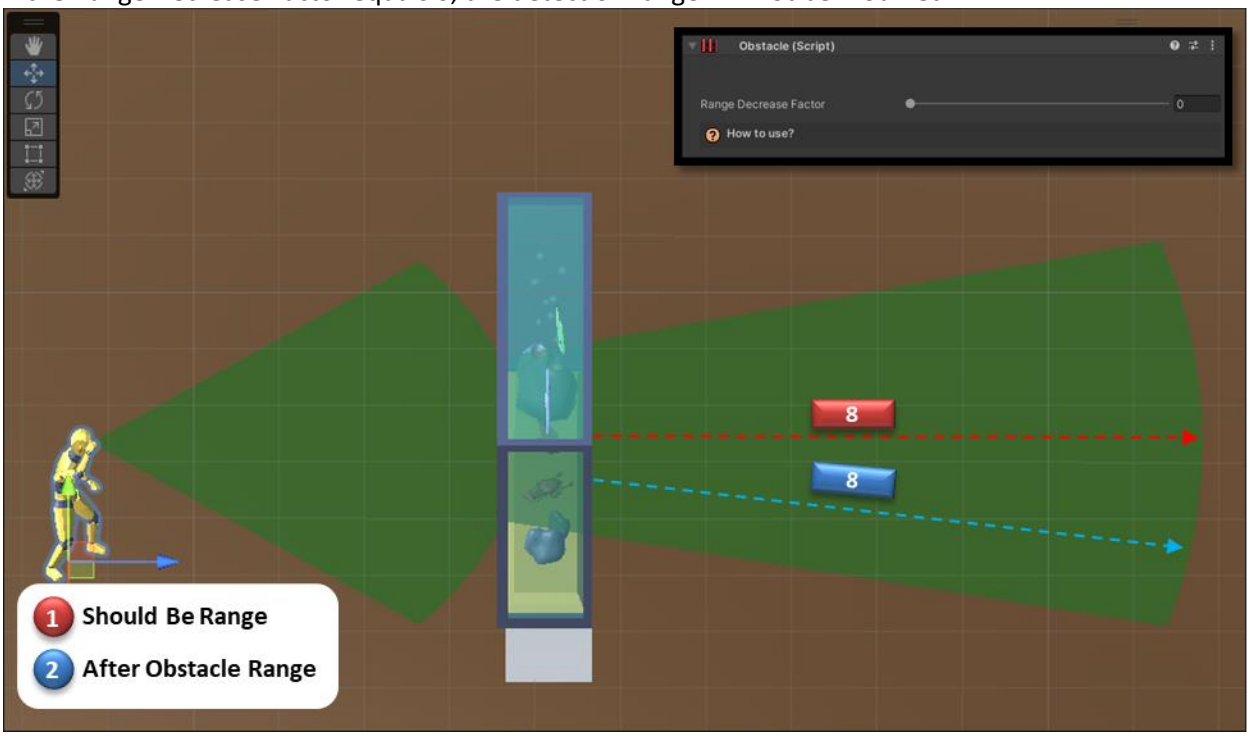
Public Methods

`Set_RangeDecreaseFactor`	`Sets float value of Range_DecreaseFactor`

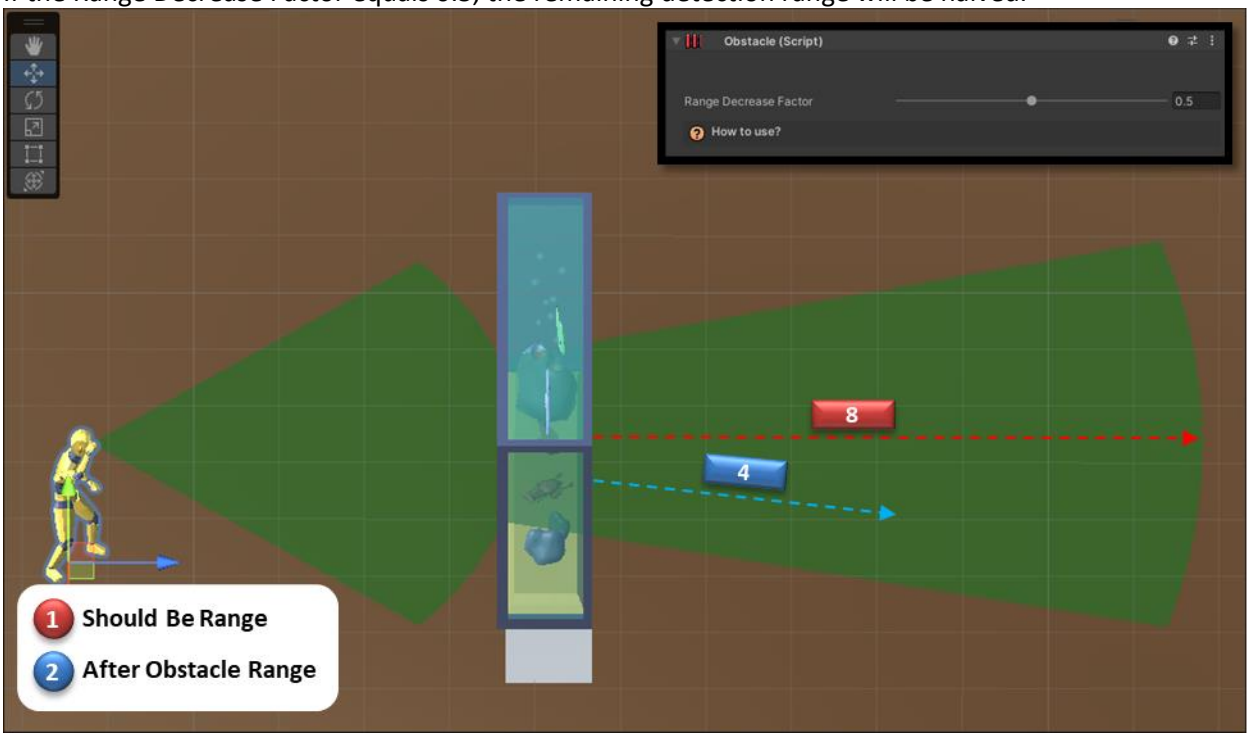
Range Decrease Factor

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:

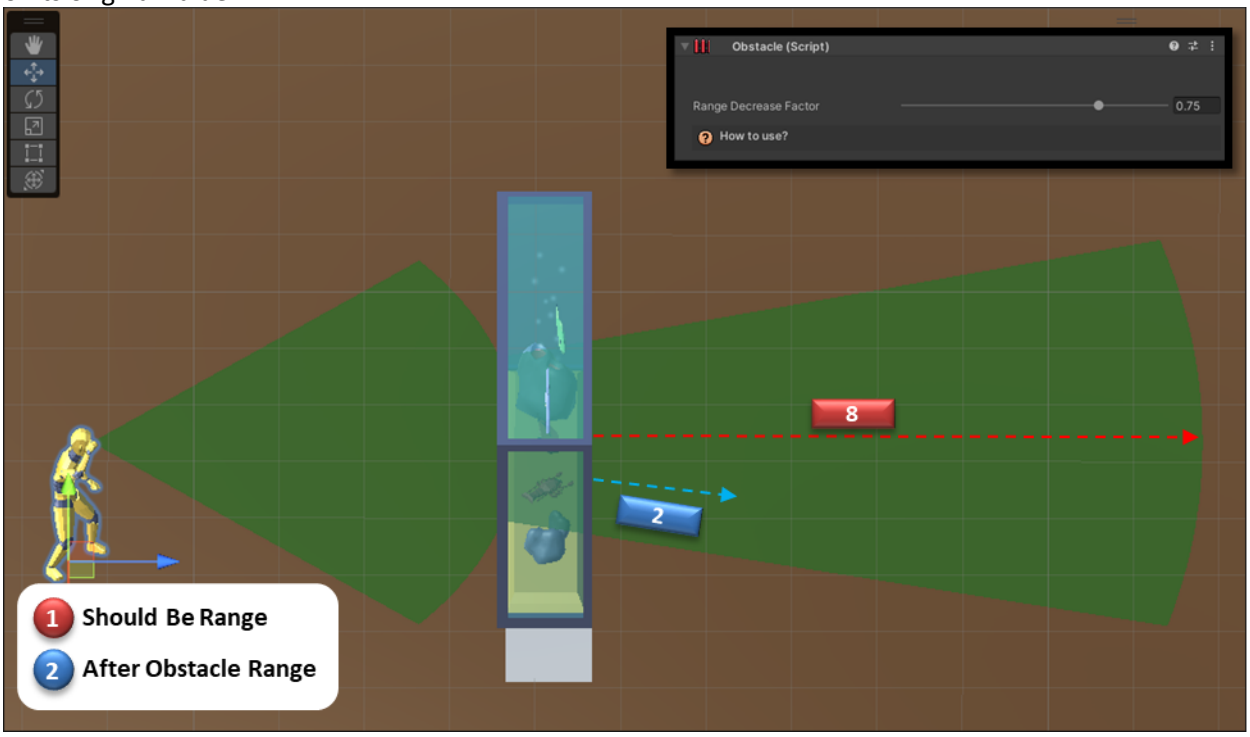
If the Range Decrease Factor equals 0, the detection range will not be modified.



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



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

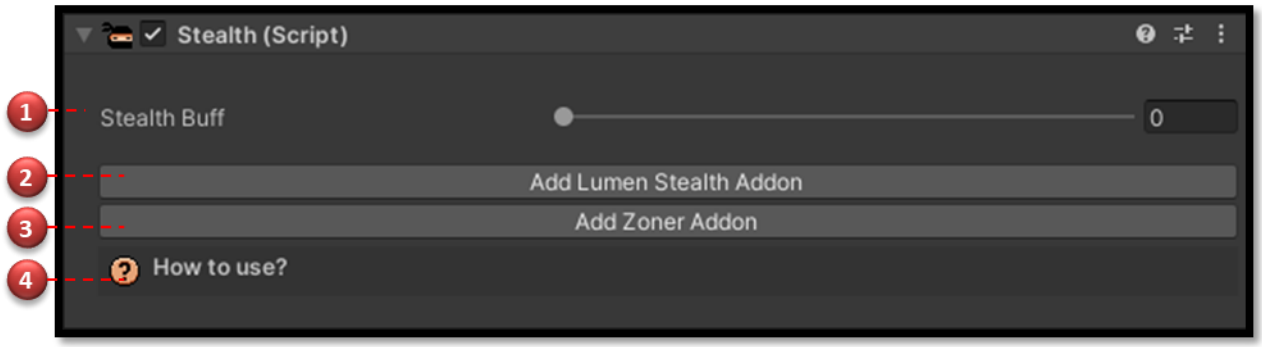


Stealth Component

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.



1	Stealth Buff	Set float value of Stealth_Buff.
2	Add Lumen Stealth Addon Button	Allows enabling Lumen Stealth Addon.
3	Add Zoner Stealth Addon Button	Allows enabling Zoner Stealth Addon.
4	'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_CustomRefreshRate	Returns true if Use_CustomRefreshRate.
Cooldown_ZonerStealthAddon	Get float value of Cooldown_ZonerStealthAddon.
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.

Public Methods

Set_StealthBuff	Set float value of Stealth_Buff.

Set_OffsetAxisY	Set float value of Set_OffsetAxisY.
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	Returns float value of Zoner_StealthAddon, returns 0 if Stealth component is not enabled;
Set_UseCustomRefreshRate	Set status of Use_CustomRefreshRate;
Set_CooldownZonerStealthAddon	Sets float value of Cooldown_ZonerStealthAddon;
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 ()

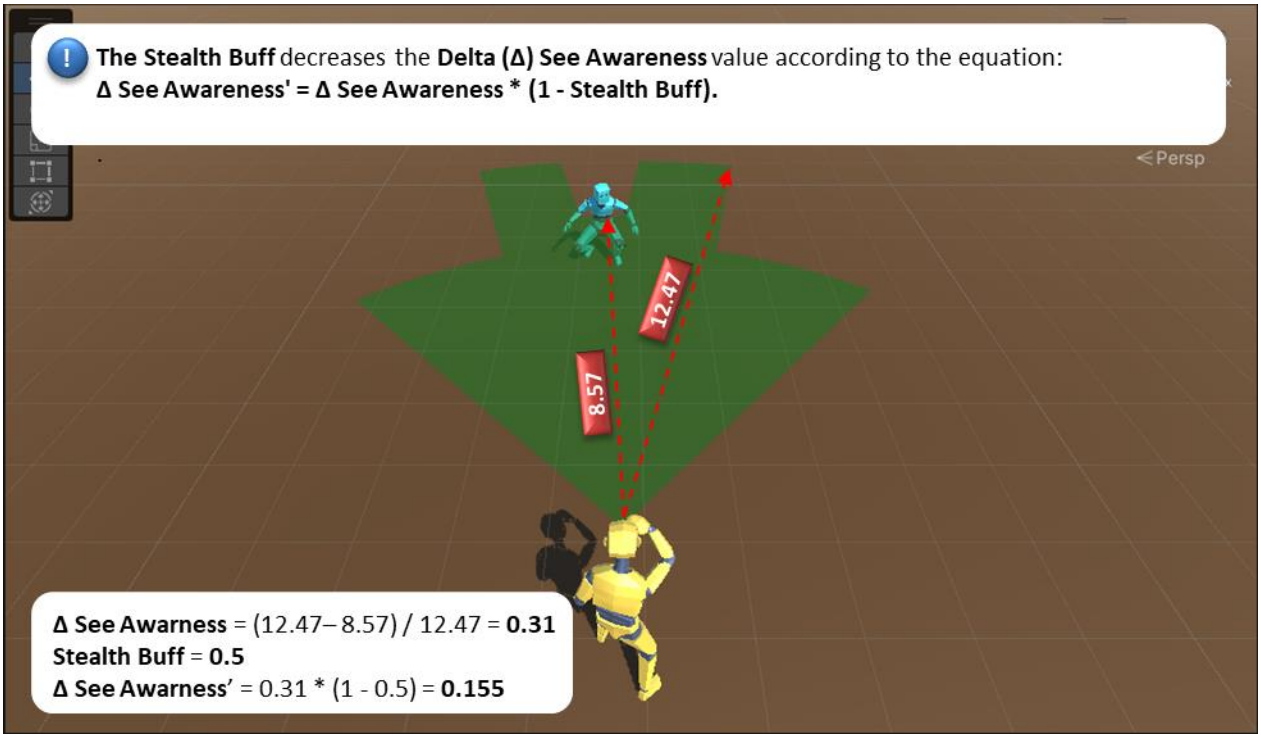
Returns the float value of the sum of:

- StealthBuff
- Lumen_StealthAddon
- Zoner_StealthAddon

If the sum exceeds 1, it will be capped at 1.

If Stealth Component is not enabled will return 0;

Reduction of the delta See Awareness



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**

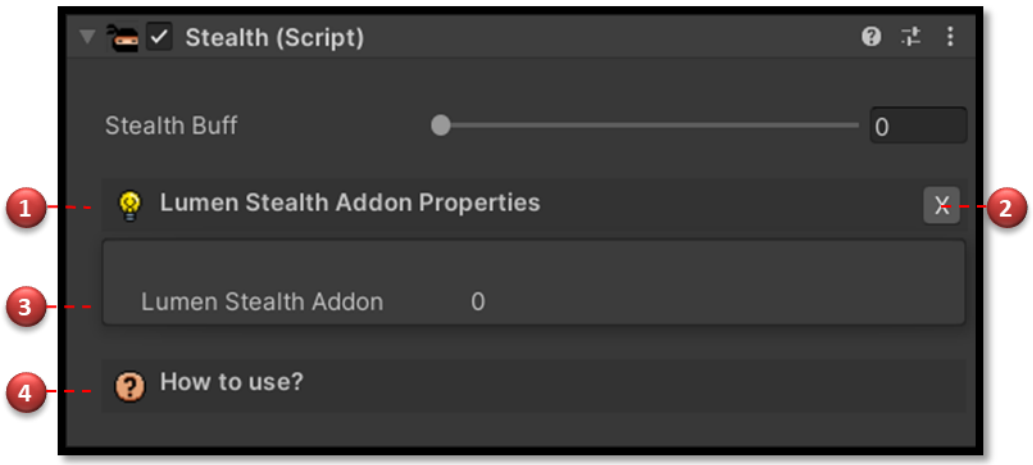
Lumen - Stealth Component Add-on

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**.



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	'How to use?' Button	Displays Help Information.

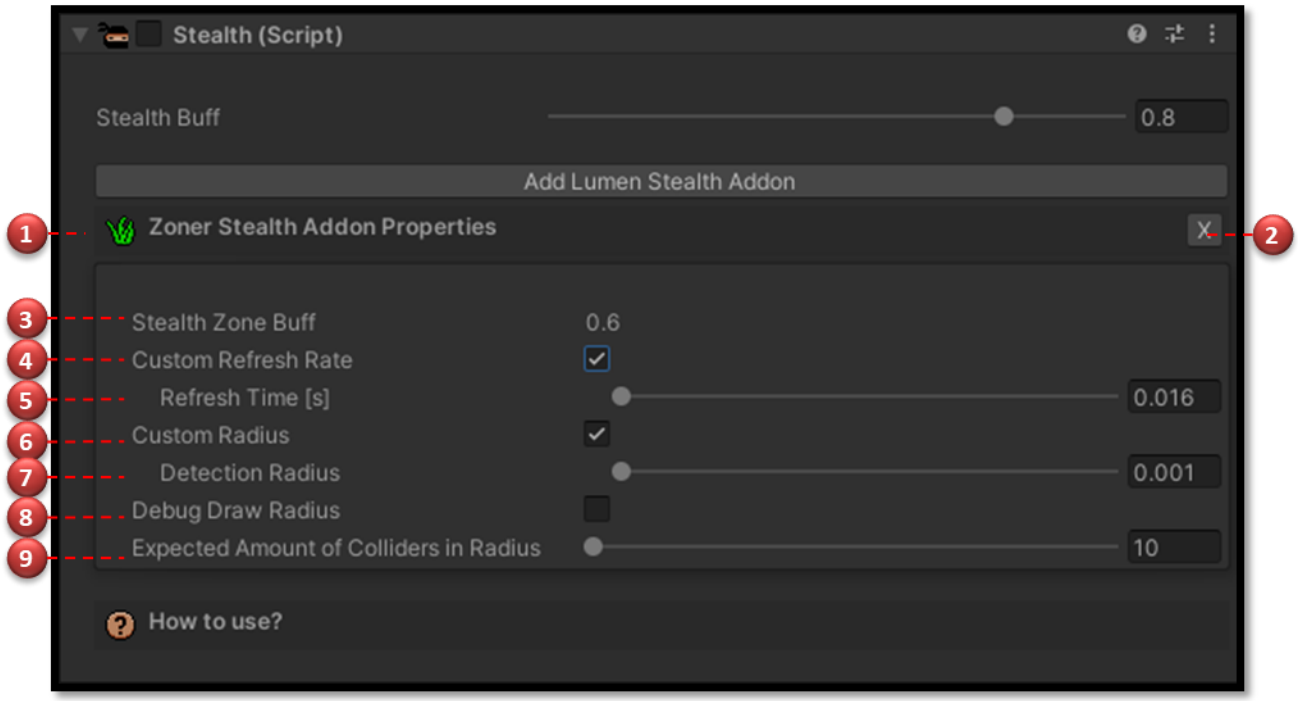
When Lumen Stealth Addon is enabled Provide_TotalStealthBuff() value will be changed while in play mode.

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.



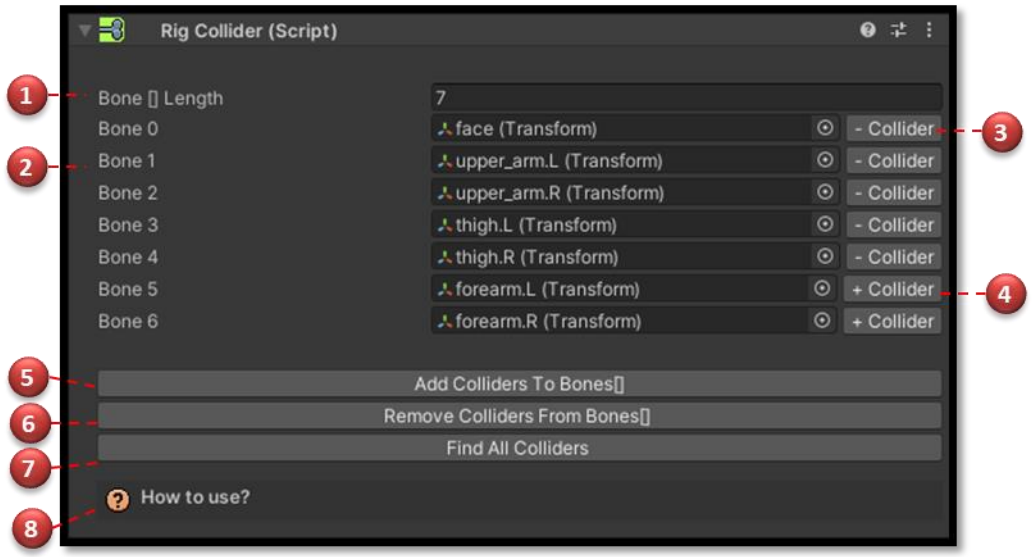
1	Zoner Stealth Addon Properties Foldout	Allows showing / hiding of Zoner Stealth Addon properties.
2	Remove Button	Allows disabling Zoner Stealth Addon.
3	Zoner Stealth Buff	Displays current Zoner Stealth Addon Value (*only during Play Mode)
4	Custom Refresh Rate	When enabled, allows determine refresh rate of checking area around for colliders with StealthZones. When disabled refresh will be conducted during each Update iteration.
5	Refresh Time [s]	Allows setting up float value of custom refresh time in seconds.
6	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.
7	Detection Radius	Allows setting up float value of radius for checking area around for colliders with SteathZones.
8	Debug Draw Radius	When enabled, with gizmos turn on, allows for graphic

		presentation of Radius_ZonerSteathAddon.
9	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.

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.



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

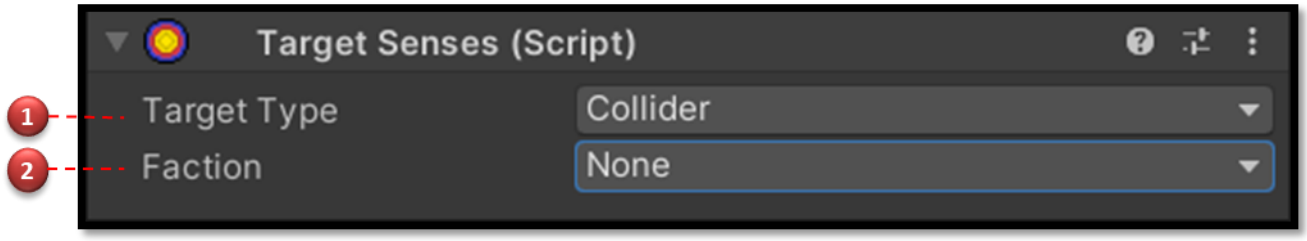
TargetSenses Component

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 **Rigidbody** component to be attached.

Allows usage of faction system (introduced in 1.02).

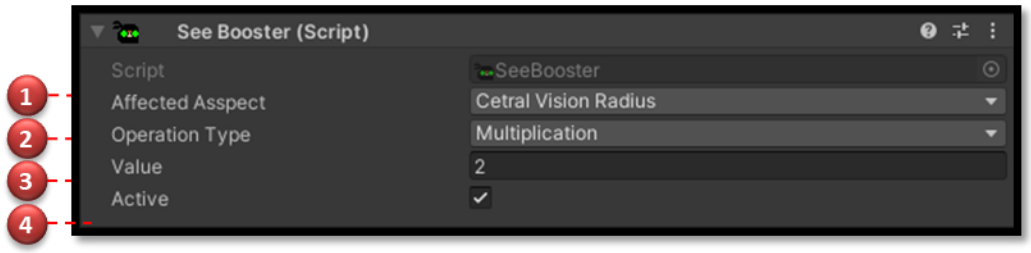


1	Target Type	Allows setting up target type. Chose 'Collider' (performance weiss solution) or 'RigCollider'.
2	Faction	Allows choosing faction. If not needed, leave as 'None'. If need to add more Factions or change names edit FactionNames.cs file.

Senses 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.



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”.

Key Methods

```
public bool TagetSensesWasDetected(
                                    int _requiredAwareness,
                                    out TargetSenses _targetSenses,
                                    FactionNames[] _factionNames,
                                    LookFor _searchType = LookFor.FirstDetected,
                                    bool _needToBeInCentralRange = true)
```

Description

Returns true if search requirements were match. Provides TargetSenses as out parameter.

In (search requirements)

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

Out

1	_targetSenses	Returns TargetSenses which meets search requirements.

```
public bool TagetSensesWasDetected(
                                    int _requiredAwareness,
                                    out TargetSenses _targetSenses,
                                    FactionNames _factionNames = FactionNames.None,
                                    LookFor _searchType = LookFor.FirstDetected,
                                    bool _needToBeInCentralRange = true)
```

Description

Returns true if search requirements were match. Provides TargetSenses as out parameter.

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	_searchType	Allows for choosing required LookFor. By default set

		as LookFor.FirstDetected.
4	_needToBeInCentralRange	Allows for choosing did at moment of request TargetSenses had to be in CentralVisionRange. By default set as true.

Out

1	_targetSenses	Returns TargetSenses which meets search requirements.

```
public bool TagetSensesWasDetected(
                                    int _requiredAwareness,
                                    FactionNames _factionNames = FactionNames.None,
                                    bool _needToBeInCentralRange = true)
```

Description

Returns true if search requirements were match.

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 bool WasDetected(
                        TargetSenses _targetSenses
                        float _requiredAwareness,
                        bool _needToBeInCentralRange = true)
```

Description

Returns true if TargetSenses provided by user was detected.

In (search requirements)

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

```
public bool WasDetected(
                        TargetSenses _targetSenses
                        float _requiredAwareness,
                        out float _awareness_See,
                        out float _awareness_Hear,
                        bool _needToBeInCentralRange = true)
```

Description

Returns true if TargetSenses provided by user was detected.

In (search requirements)

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

Out

1	_awareness_See	Returns float value of See Awareness of checked TargetSenses.
2	_awareness_Hear	Returns float value of Hear Awareness of checked TargetSenses.

```
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.