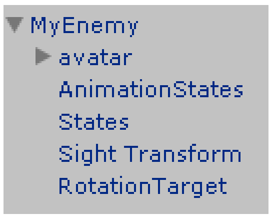
INDEX

1. Setting up AI Behavior

1.1 Components

First thing to do is to create an empty GameObject and add your character/avatar to it.

On this GameObject add the "AIBehaviors" script and put the "NavMeshAgent" (you will see some extra GameObjects and Components be added automatically).



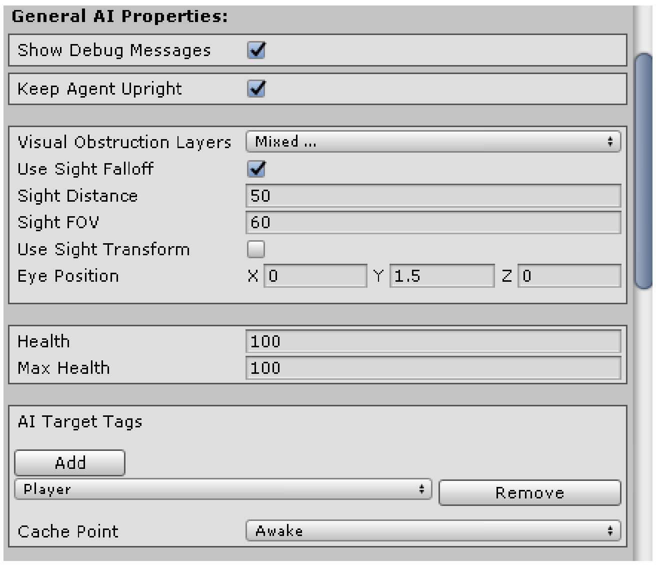
```
▼ MyEnemy
    ▶ avatar
      AnimationStates
      States
      Sight Transform
      RotationTarget
```

You should also add to this GameObject the "CharacterAnimator" or "MecanimAnimation" script, depending if you use Legacy or Mecanim animations.

In the inspector you can set up all the different states that your character will use, and the conditions and triggers for each of them.

1.2 General AI Properties

This are settings applied to all the AI. Here you can set the sight properties of the AI, the health, and the target tag/s (this will be the most common target tag, but you can also set specific tags for each state and each trigger).



General AI Properties:

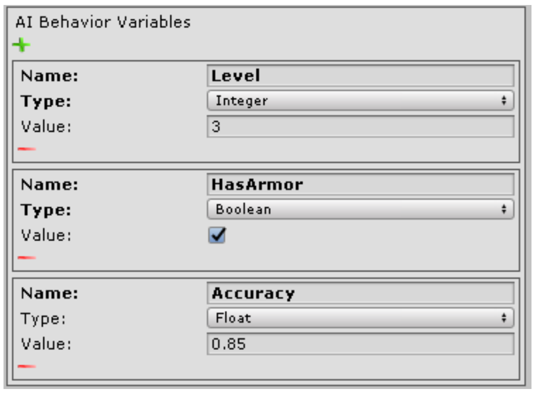
Show Debug Messages	✓
Keep Agent Upright	✓
Visual Obstruction Layers	Mixed ...
Use Sight Falloff	✓
Sight Distance	50
Sight FOV	60
Use Sight Transform	☐
Eye Position	X 0 Y 1.5 Z 0
Health	100
Max Health	100

AI Target Tags

Player

Cache Point: Awake

You can also add custom variables that you can use in game with the **CheckVariableTrigger**, and that you can change when entering a specific state. Allowed types are: Floats, Integers and Booleans.



AI Behavior Variables

Name:	**Level**
Type:	Integer
Value:	3

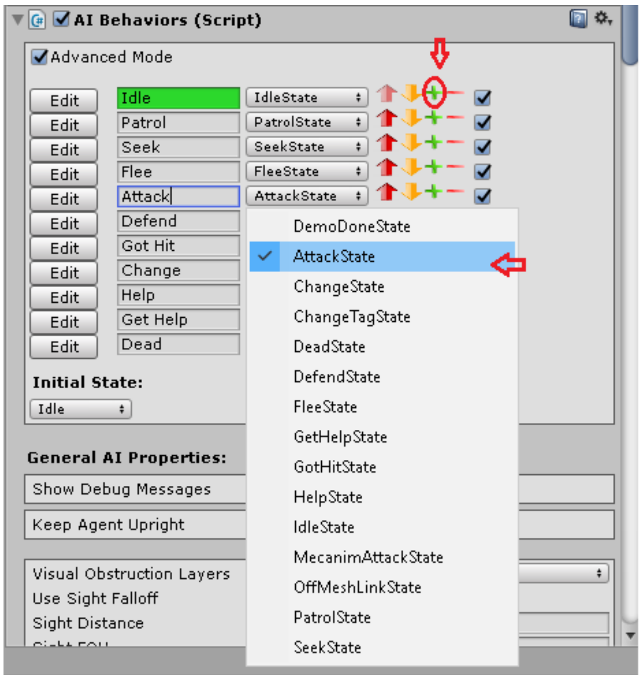
Name:	**HasArmor**
Type:	Boolean
Value:	✓

Name:	**Accuracy**
Type:	Float
Value:	0.85

1.3 Adding and Setting up States

To add a state just **click the green "+" button** near one of the states and **choose** from the list what **kind of state** it is. Then for each state set the properties and the triggers to change between states.

Note: This states are scripts, if you want a custom one, you can write your own (extending the BaseState) and it will appear on this list.



AI Behaviors (Script)

✓ Advanced Mode

Edit	Idle	IdleState	✓
Edit	Patrol	PatrolState	✓
Edit	Seek	SeekState	✓
Edit	Flee	FleeState	✓
Edit	Attack	AttackState	✓
Edit	Defend		
Edit	Got Hit		
Edit	Change		
Edit	Help		
Edit	Get Help		
Edit	Dead		

Initial State: Idle

General AI Properties:

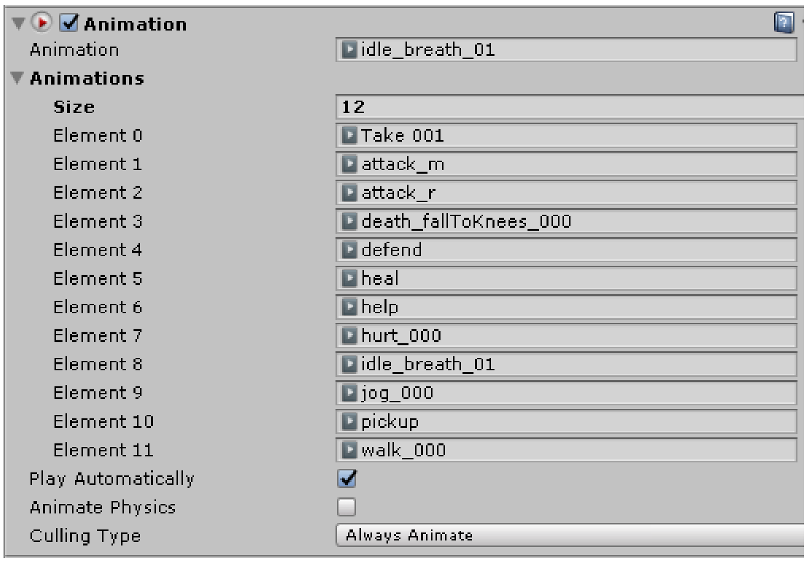
1.4 Animations

To set up the animations you have to add them to the "AIAnimationStates" script (This script is automatically added with the AIBeahaviors).

Depending if you use Mecanim or Legacy animations it will be different.

1.4.1 Legacy animations

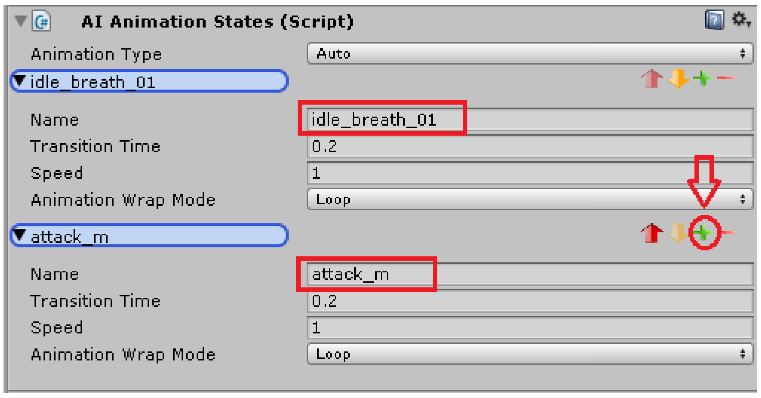
With **Legacy,** you will need to add the **"Animation"** component to your **character/avatar** and add to it all your animations.



Animation

Animation	idle_breath_01
Animations	
Size	12
Element 0	Take 001
Element 1	attack_m
Element 2	attack_r
Element 3	death_fallToKnees_000
Element 4	defend
Element 5	heal
Element 6	help
Element 7	hurt_000
Element 8	idle_breath_01
Element 9	jog_000
Element 10	pickup
Element 11	walk_000
Play Automatically	✓
Animate Physics	
Culling Type	Always Animate

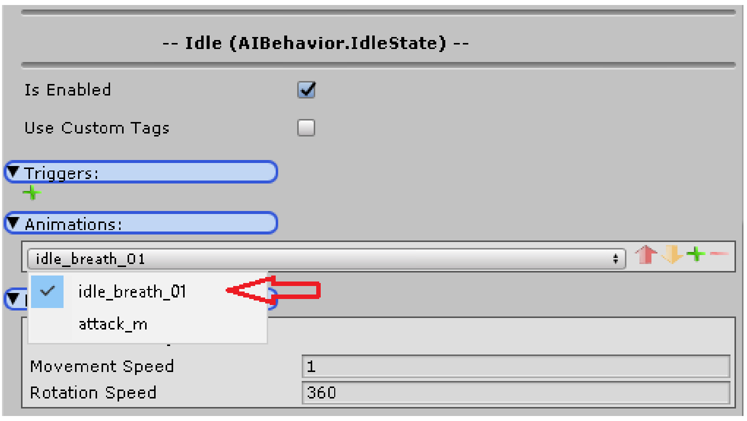
Then go back to the main GameObject and add the animations to the "AIAnimationStates" script, "+" button, and using the same names as the animations.



AI Animation States (Script)

Animation Type	Auto
idle_breath_01	
Name	idle_breath_01
Transition Time	0.2
Speed	1
Animation Wrap Mode	Loop
attack_m	
Name	attack_m
Transition Time	0.2
Speed	1
Animation Wrap Mode	Loop

Finally in each state pick which animation it has to use, you can have more than one animation, in that case it will choose a random one each time it enters the state.


-- Idle (AIBehavior.IdleState) --

Is Enabled ✓

Use Custom Tags ☐

Triggers:

Animations:

idle_breath_01

✓ idle_breath_01

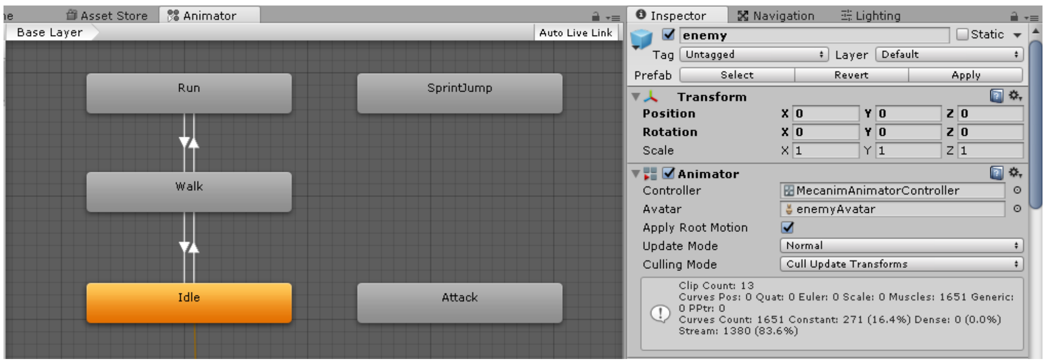
attack_m

Movement Speed 1

Rotation Speed 360

1.4.2 Mecanim animations

With **Mecanim,** you will need to add the **"Animator"** component to your **character/avatar** with the avatar of your character and a controller.



enemy ☐ Static

Tag Untagged Layer Default

Prefab Select Revert Apply

Transform

Position	X	0	Y	0	Z	0
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

✓ Animator

Controller MecanimAnimatorController

Avatar enemyAvatar

Apply Root Motion ✓

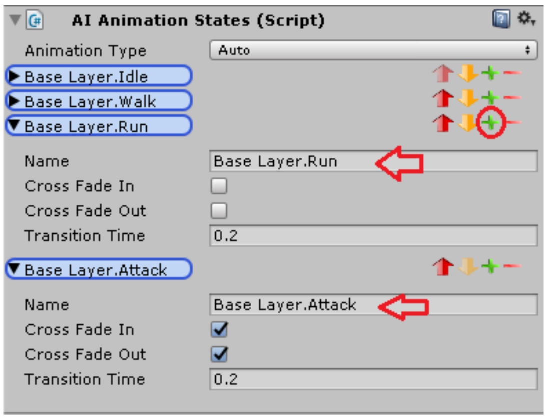
Update Mode Normal

Culling Mode Cull Update Transforms

```
Clip Count: 13
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 1651 Generic:
0 PPtr: 0
Curves Count: 1651 Constant: 271 (16.4%) Dense: 0 (0.0%)
Stream: 1380 (83.6%)
```

In this controller you have "layers" and "states" in each layer. To use a state from the AIBehavior script you have to add the animations to the "AIAnimationStates" script, "+" button, and specifying the layer and the name of the state: "layer.state".

The "cross fade in" and "cross fade out" options should be checked if you want the AI to have full control over the animations without any additional code.


AI Animation States (Script)

Animation Type Auto

Base Layer.Idle

Base Layer.Walk

Base Layer.Run

Name Base Layer.Run

Cross Fade In ☐

Cross Fade Out ☐

Transition Time 0.2

Base Layer.Attack

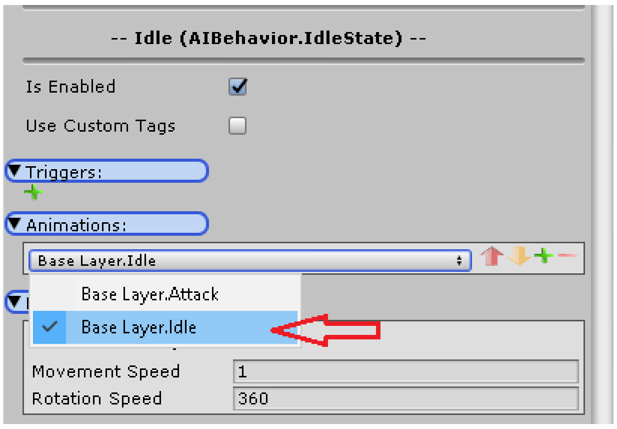
Name Base Layer.Attack

Cross Fade In ✓

Cross Fade Out ✓

Transition Time 0.2

Finally in each state pick which animation it has to use



-- Idle (AIBehavior.IdleState) --

Is Enabled	✓
Use Custom Tags	☐

Triggers:

Animations:

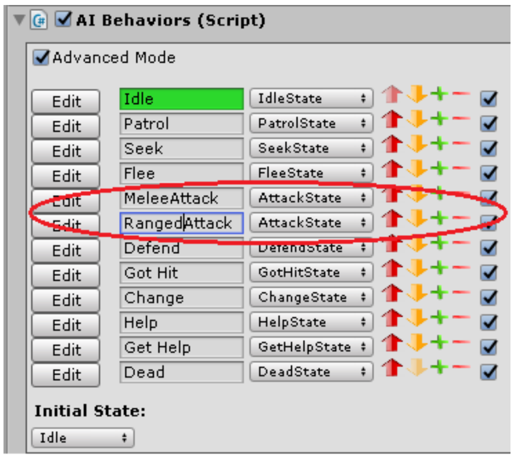
Base Layer.Idle

- Base Layer.Attack
- ✓ Base Layer.Idle

Movement Speed	1
Rotation Speed	360

2. States

The states handle the different behaviors of your AI. You can have so many states as you want, and you can also have several states of the same kind (for example: 2 AttackStates)



AI Behaviors (Script)

✓ Advanced Mode

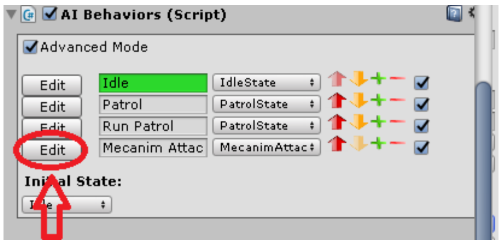
Edit	Idle	IdleState	✓
Edit	Patrol	PatrolState	✓
Edit	Seek	SeekState	✓
Edit	Flee	FleeState	✓
Edit	MeleeAttack	AttackState	✓
Edit	RangedAttack	AttackState	✓
Edit	Defend	DefendState	✓
Edit	Got Hit	GotHitState	✓
Edit	Change	ChangeState	✓
Edit	Help	HelpState	✓
Edit	Get Help	GetHelpState	✓
Edit	Dead	DeadState	✓

Initial State:

Idle

2.1 State Properties

To set the properties for each state just click the "Edit" button on the left of the state you want to edit, and scroll down in the inspector to change the settings. (Depending on the state it will have more or less properties to modify)



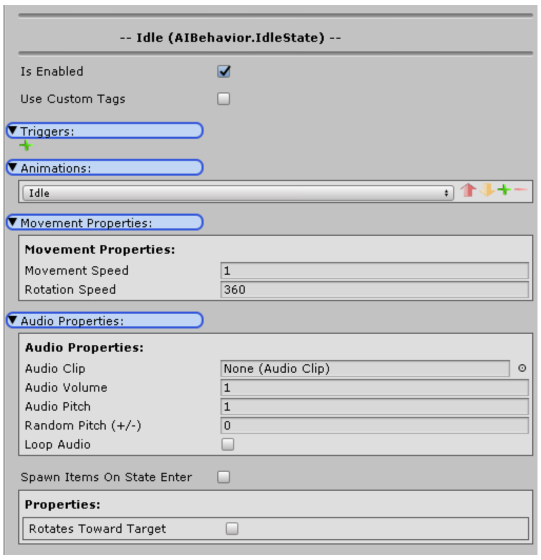
AI Behaviors (Script)

✓ Advanced Mode

Edit	Idle	IdleState	✓
Edit	Patrol	PatrolState	✓
Edit	Run Patrol	PatrolState	✓
Edit	Mecanim Attac	MecanimAttac	✓

Initial State:

Idle


-- Idle (AIBehavior.IdleState) --

Is Enabled ✓

Use Custom Tags

Triggers:

Animations:

Idle

Movement Properties:

Movement Properties:

Movement Speed	1
Rotation Speed	360

Audio Properties:

Audio Properties:

Audio Clip	None (Audio Clip)
Audio Volume	1
Audio Pitch	1
Random Pitch (+/-)	0
Loop Audio	

Spawn Items On State Enter

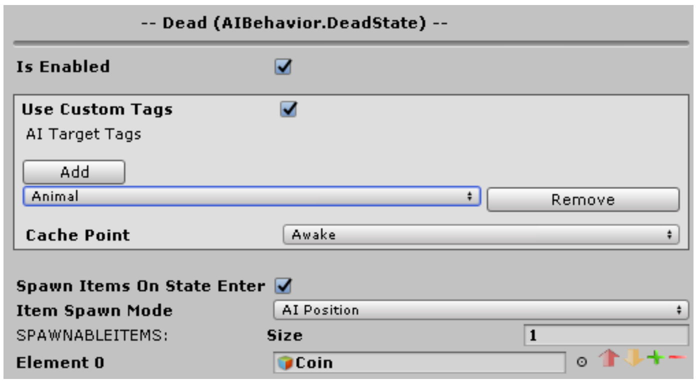
Properties:

Rotates Toward Target

2.2 General State Properties

There are some settings that are common to all the states.

- **Is Enabled:** Enables or disables the state (if a state is disabled the AI can't use it)
- **Use Custom Tags:** Tags that will be used for the state and all the triggers of the state except the ones that have their own custom tag. (if you don't check it, the AI will use the general AI target tag/s).


-- Dead (AIBehavior.DeadState) --

Is Enabled ✓

Use Custom Tags ✓

AI Target Tags

Add

Animal | Remove

Cache Point: Awake

Spawn Items On State Enter ✓

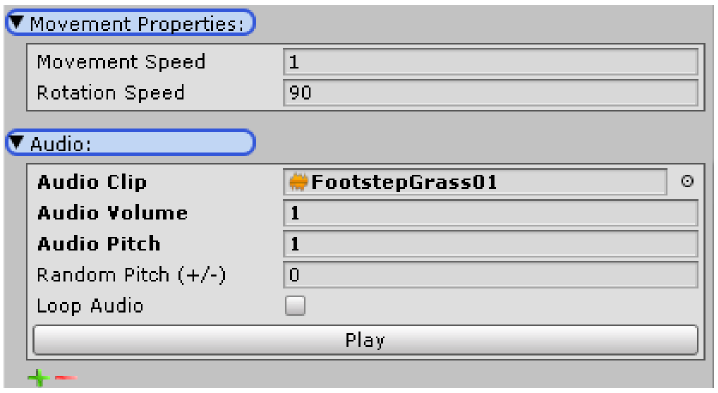
Item Spawn Mode: AI Position

SPAWNABLEITEMS: **Size** 1

Element 0: Coin

- **Triggers** (specified on Point 3).
- **Animations** (specified on Point 1.4).
- **Movement Properties:** Specify how fast the AI will move and rotate in that specific state (remember setting the "movement speed" to 0 in states where the AI shouldn't move).
- **Audio Properties:** You can add a sound to be played when entering the state, if you add more than one it will pick one randomly.
- **Item Spawning Properties:** Allows you to spawns one or more items at the specified place when entering the state. The "Item Spawn Mode" options are: At AI position, at specific coordinates or passing a transform (like a spawn point).
- **Changing Variable Properties:** Allows you to change the value of the custom variables (specified on Point 1.2), you have the option "Change To" to set a specific value to the

variable and "Increase by" and "Decrease by" (only for ints and floats). The changes will apply when entering the state.

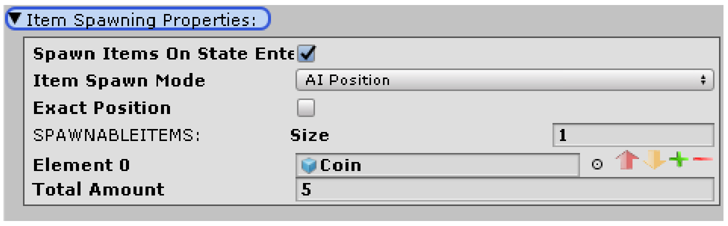


Movement Properties:

Movement Speed	1
Rotation Speed	90

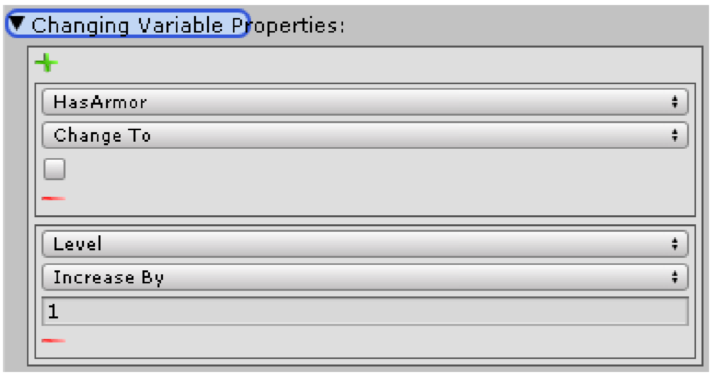
Audio:

Audio Clip	**FootstepGrass01**
Audio Volume	**1**
Audio Pitch	**1**
Random Pitch (+/-)	0
Loop Audio	☐



Item Spawning Properties:

Spawn Items On State Ente	✓	
Item Spawn Mode	AI Position	
Exact Position	☐	
SPAWNABLEITEMS:	**Size**	**1**
Element 0	**Coin**	
Total Amount	**5**	

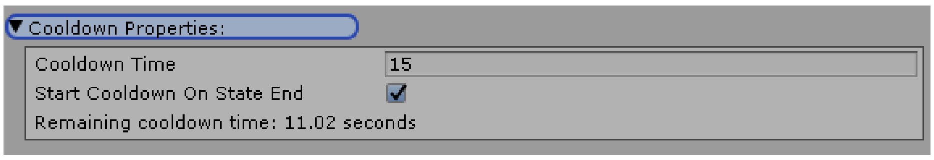


Changing Variable Properties:

HasArmor
Change To
☐

Level
Increase By
1

Finally some states are CooldownableStates and have **Cooldown Properties.** This states can have a "**Cooldown Time**" that sets how long the state will be unavaiable. The time will start to count when the state ends if you check "**Start Cooldown On State End**" if not, it will start at entering the state. A trigger won't enter a state if it's cooling down even if the condition is matched.



Cooldown Properties:

Cooldown Time	15
Start Cooldown On State End	✓

Remaining cooldown time: 11.02 seconds

2.3 Types of states

2.3.1 Idle

The idle state don't needs much explanation, the AI just stays in his position. The only option to check is "Rotate Towards Target" that makes the AI be facing the player/target all the time.

2.3.2 Patrol State

The patrol state is for making your AI to walk following a specific path. "**Patrol mode**" allows you to choose between 4 different types of patrolling:

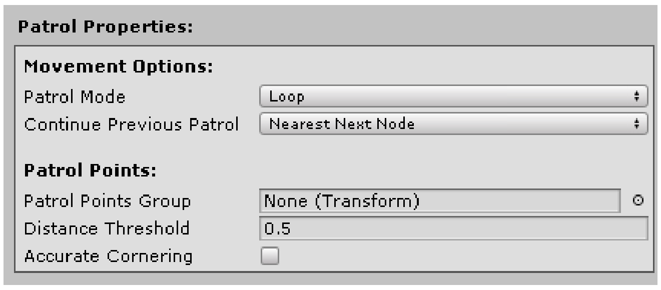
- **Once**: Does the patrol just once and then changes to another state "End Patrol Transition".
- **Loop**: Keeps patrolling all the time until a trigger makes him change to another state.
- **Ping Pong**: Will do the patrol in one way and when it finishes it will do it the other way.
- **Random**: Patrols randomly through the points

The "**Continue Previous Patrol**" is to pick how the AI should start/continue patrolling.

- **Reset**: Starts at the first point
- **Continue Previous:** Continues with the last point it was going to.
- **Nearest Node**: Picks the closest point to the AI.
- **Nearest Next Node**: Picks the point after the closest one.
- **Random**: Just picks a random point.

You will have to group all the points where you want your AI to patrol under one GameObject and drag it to "**Patrol Points Group**". Set also a "**Distance Threshold**" according how close you want the AI to get to the points.

Finally there is the "**Accurate Cornering**" option, only use it if you don't like how your AI is passing through the points or if it's missing them (for example if your AI moves very fast). Then you have to set the "**Cornering Error**" value (usually 1 is fine).


Patrol Properties:

Movement Options:

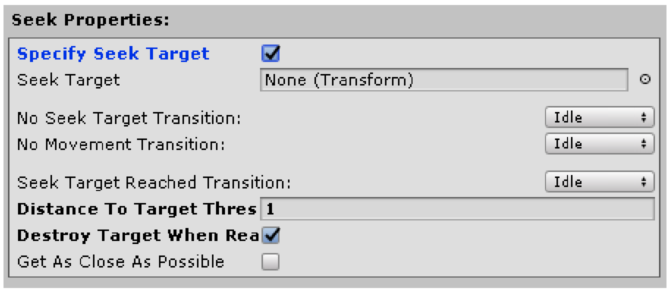
Patrol Mode	Loop
Continue Previous Patrol	Nearest Next Node

Patrol Points:

Patrol Points Group	None (Transform)
Distance Threshold	0.5
Accurate Cornering	☐

2.3.3 Seek State

The seek state can be used for different purposes, as chasing the player to attack him or just follow him, seek objects, go to a specific point, etc. You can set a specific point where it has to go checking "**Specify Seek Target**" and dragging the point to "**Seek Target**". If you don't pick that option the AI will look for objects with the target tag/s. "**No Seek Target Transition**" is the state it will change in if there is no target. "**No Movement Transition**" is the state it will go into when the AI stops for whatever reason (for example, it can't get to the target). "**Seek Target Reached Transition**" is the state it will change in when it reaches the target (for example the AI is chasing the Player and when it reaches him it changes to an AttackState). The "**Distance To Target Threshold**" is how close the AI gets to the target. You can also enable "**Destroy Target When Reached**" that will destroy the target GameObject. Finally the "**Get As Close As Possible**" option is in case that the AI can't get to the target it will at least approach as much he can.



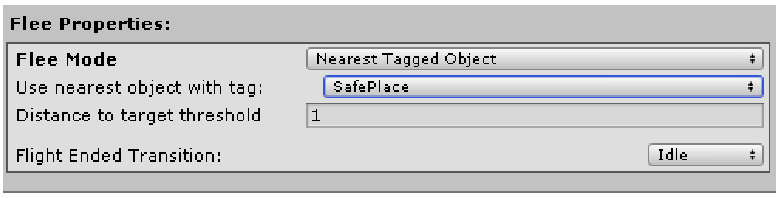
Seek Properties:	
Specify Seek Target	✓
Seek Target	None (Transform)
No Seek Target Transition:	Idle
No Movement Transition:	Idle
Seek Target Reached Transition:	Idle
Distance To Target Thres	1
Destroy Target When Rea	✓
Get As Close As Possible	

2.3.4 Flee State

The flee state is use to run away from the player or something else. The "**Flee Mode**" is the way it will flee:

- **Nearest Tagged Object**: This will use the "**nearest object with tag**" objects as refuges.
- **Fixed Target**: Is to pass a Transform of the place it has to go
- **Direction**: Is to run in a specific direction, no matter where the player/object is.
- **Away from Nearest Tagged Object:** This will make the AI run in the opposite direction.

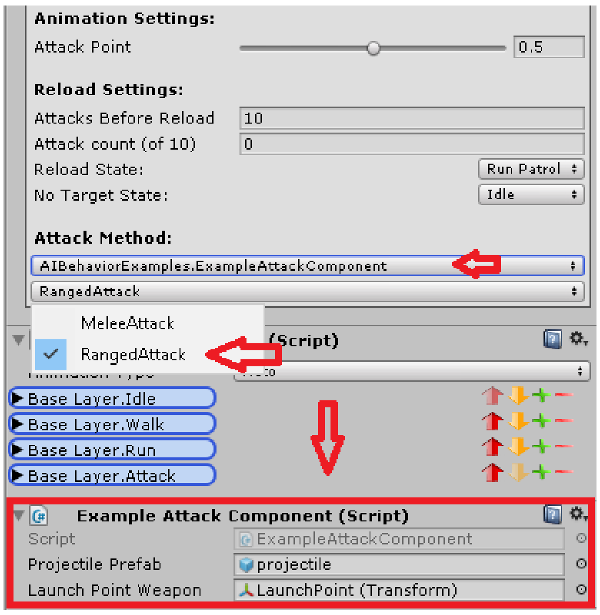
The "**Distance to target threshold**" is how close it will get to the "safe place", or in case of specific direction and running away you will have to set the "**Stop Flee Distance**". Finally the "**Flight Ended Transition**" is the state it will go into when the flight is finished.



Flee Properties:	
Flee Mode	Nearest Tagged Object
Use nearest object with tag:	SafePlace
Distance to target threshold	1
Flight Ended Transition:	Idle

2.3.5 Attack State & Mecanim Attack State

The attack state needs an attack script to know what to do when the AI attacks. You can use the "ExampleAttackComponent" or you can write your own. Add it to the root GameObject and then in the AttackState properties select the method to use when attacking.

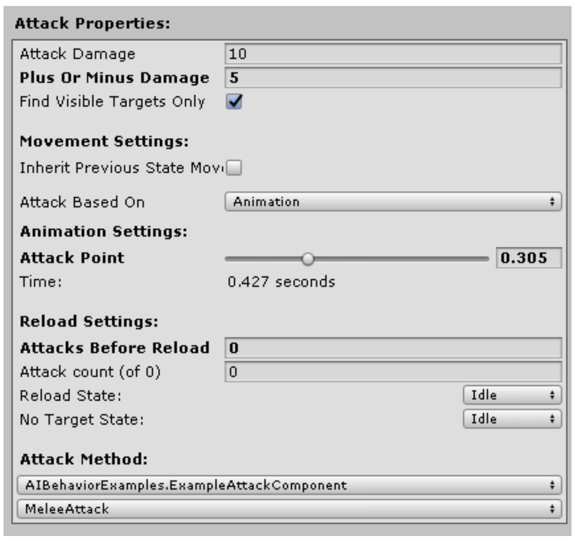

Animation Settings:
Attack Point 0.5

Reload Settings:
Attacks Before Reload 10
Attack count (of 10) 0
Reload State: Run Patrol
No Target State: Idle

Attack Method:
AIBehaviorExamples.ExampleAttackComponent
RangedAttack

Example Attack Component (Script)
Script ExampleAttackComponent
Projectile Prefab projectile
Launch Point Weapon LaunchPoint (Transform)

In the attack properties you can set how much the base "**Attack Damage**" is, and add some randomness to the damage with "**Plus Or Minus Damage**", set if it will only "**Find Visible Targets Only**" (sometimes it's better to disable it, as in melee attacks). "**Inherit Previous State Movement**" means that the AI will keep moving as the previous state tells him to do while attacking (for example keep patrolling or seeking), "**Attack Based On**" defines at what moment the "**Attack Method**" will be called, it can be based on a specific point of the animation or with a time interval. The "**Attacks Before Reload**" is the amount of times the attack method will be called before changing to the reload state. "**Attack count**" is just the current count of attacks. The "**Reload State**" is the state it goes in when the attack limit is reached. And the "**No Target State**" is the state it changes into when it couldn't find a target.


Attack Properties:
Attack Damage 10
Plus Or Minus Damage 5
Find Visible Targets Only ✓

Movement Settings:
Inherit Previous State Movi ☐

Attack Based On Animation

Animation Settings:
Attack Point 0.305
Time: 0.427 seconds

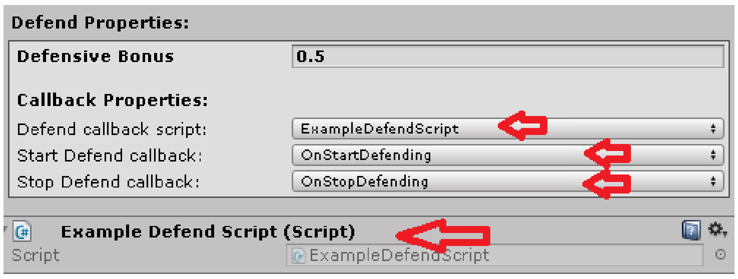
Reload Settings:
Attacks Before Reload 0
Attack count (of 0) 0
Reload State: Idle
No Target State: Idle

Attack Method:
AIBehaviorExamples.ExampleAttackComponent
MeleeAttack

The MecanimAttackState is almost the same as the AttackState it just extends it to use mecanim animation.

2.3.6 Defend State

The defend state is to have the AI in a defend position and have a "**Defensive Bonus**" for the damage calculation. It needs a defend script with the methods to call when it starts and ends defending, you can use the "ExampleDefendScript" or write your own (This script just replaces the damage multiplier by the defensive bonus value while its defending).



Defend Properties:

Defensive Bonus	0.5

Callback Properties:

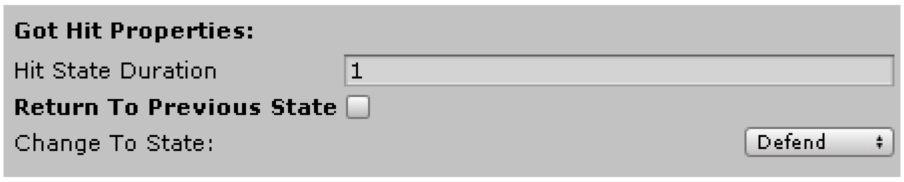
Defend callback script:	ExampleDefendScript
Start Defend callback:	OnStartDefending
Stop Defend callback:	OnStopDefending

Example Defend Script (Script)

Script	ExampleDefendScript

2.3.7 Got Hit State

The got hit state is just a state to go in when the AI was hit and play the animation for it. You can set how long the AI will stay in this state in "**Hit State Duration**" (usually more or less what the animation lasts). If you check "**Return To Previous State**" it will go back to the state it was before being hit, or you can choose another state in "**Change To State**".

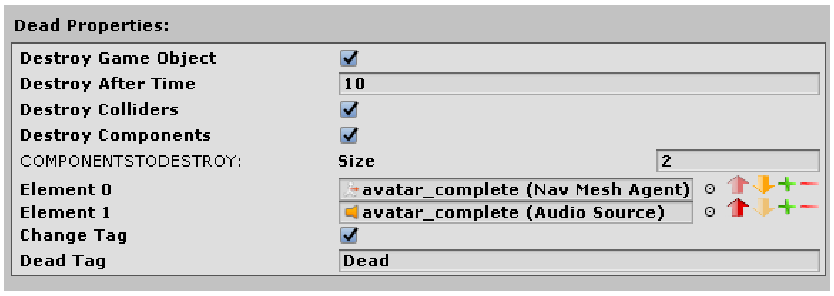


Got Hit Properties:

Hit State Duration	1
Return To Previous State	☐
Change To State:	Defend

2.3.8 Dead State

The dead state is obviously the state the AI goes into when he dies. You can check the "**Destroy Game Object**" option to destroy the GameObject after the time specified in "**Destroy After Time**", you can also check "**Destroy Colliders**" to avoid to keep hitting the AI while its dead. "**Destroy Components**" allows you to destroy other components of the AI, just drag them into the list. And finally you can change the tag of the AI when he dies checking "**Change Tag**" and setting it on "**Dead Tag**".



Dead Properties:	
Destroy Game Object	✓
Destroy After Time	10
Destroy Colliders	✓
Destroy Components	✓
COMPONENTSTODESTROY:	Size 2
Element 0	avatar_complete (Nav Mesh Agent)
Element 1	avatar_complete (Audio Source)
Change Tag	✓
Dead Tag	Dead

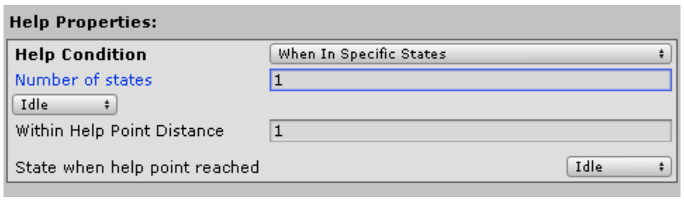
2.3.9 Get Help State

The get help state is a way that an AI can call other AIs (for example when he sees the player). There is the "**Get Help From Objects With Tag**" value that specifies the tag of the "helpers", the "**Help Radius**" value that is from how far they will come, you can specify in "**State Duration**" how long it will stay in this state and in "**Change To State**" the state it will change into after it ended.

Note: The other AIs need to have a HelpState to be able to respond to the calling.

2.3.10 Help State

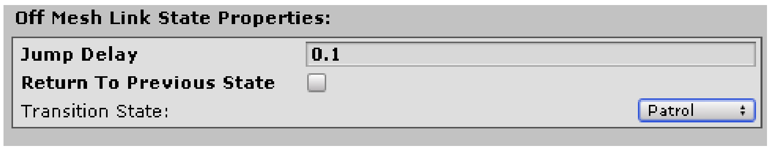
The help state is the one used for going to the position of an AI asking for help through a GetHelpState. You can pick the "**Help Condition**" that specifies if the AI will go help another always or only when he is in specific states. If you pick this second option you have to set the "**Number Of States**" and picking which ones (if the AI is in a state that is not in the list he will not go help the one who is asking for help). The "**Within Help Point Distance**" specifies how close the AI will approach to the "help point" (the position where the other AI called) and finally in "**State When Help Point Reached**" is the state it will change into when he got there.



Help Properties:	
Help Condition	When In Specific States
Number of states	1
Idle	
Within Help Point Distance	1
State when help point reached	Idle

2.3.11 Off Mesh Link State

This state is a hay to make the AI jump from an OffMeshLink, the settings for it are "**Jump Delay**" that specifies how long the AI will wait since he gets to the link until he jumps, then you can choose between checking "**Return To Previous State**" or a "**Transition State**" that sets the state the AI will change into after jumping.


Off Mesh Link State Properties:

Jump Delay	0.1
Return To Previous State	☐
Transition State:	Patrol

2.3.12 Change State

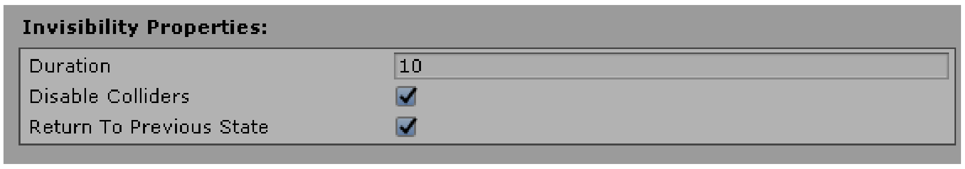
The change state is to replace the whole AI with another GameObject set in the "**Change into**" field. It will put the new GameObject in the exact same position and rotation.

2.3.13 Change Tag State

The change tag state is just a state to change the tag of the AI. The only option is "**Change To Tag**" where you can set the new tag for the AI.

2.3.14 Get Invisible State

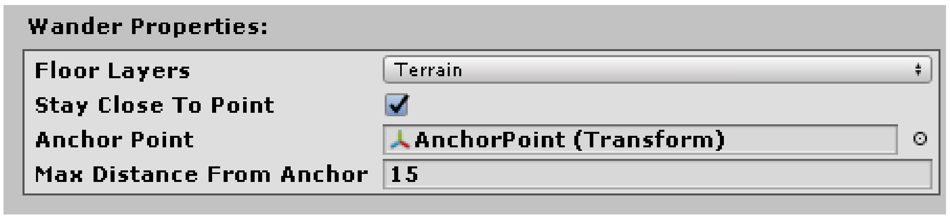
The get invisible state is a simple way to make an AI invisible for a period of time specified in "**Duration**", you can check "**Disable Colliders**" if you don't want that anyone will be able to hit him while he is invisible. The AI will not stay on this state while he is invisible, it will go back to the previous if "**Return To Previous State**" is checked or change to the specified state.


Invisibility Properties:

Duration	10
Disable Colliders	✓
Return To Previous State	✓

2.3.15 Wander State

The wander state makes the AI walk around without a concrete destination. It works through raycasts and the sight properties of the AI to look for places where he can go to. You will have to put your walkable surfaces in one or more layers and specify them in the "**Floor Layers**" property. There is also the "**Stay Close To Point**" option that allows you to set an "**Anchor Point**" that will not allow the AI go further from it than the "**Max Distance From Anchor**".


Wander Properties:

Floor Layers	Terrain
Stay Close To Point	✓
Anchor Point	AnchorPoint (Transform)
Max Distance From Anchor	15

2.3.16 Step Back State

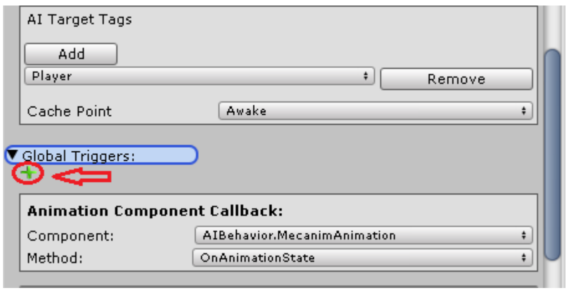
The step back state is for AIs (usually NPCs) that have to keep a distance from the player or other AIs. It will just step back facing the target. You can specify the **"distance"** you want the AI to go away from the player/target and the "**stopThreshold**" that specifies how close it will get to the point generated by the state. If you are in a terrain that is not flat you'll probably need to increase this value. When it reaches the distance it will change to the "**Next State**".

3. Triggers

3.1 Global triggers

This triggers are checked all the time, if the condition matches it will interrupt whatever state the AI is to go to the specified state. This is not as common to use as the state triggers, only use it for conditions that apply to any state.

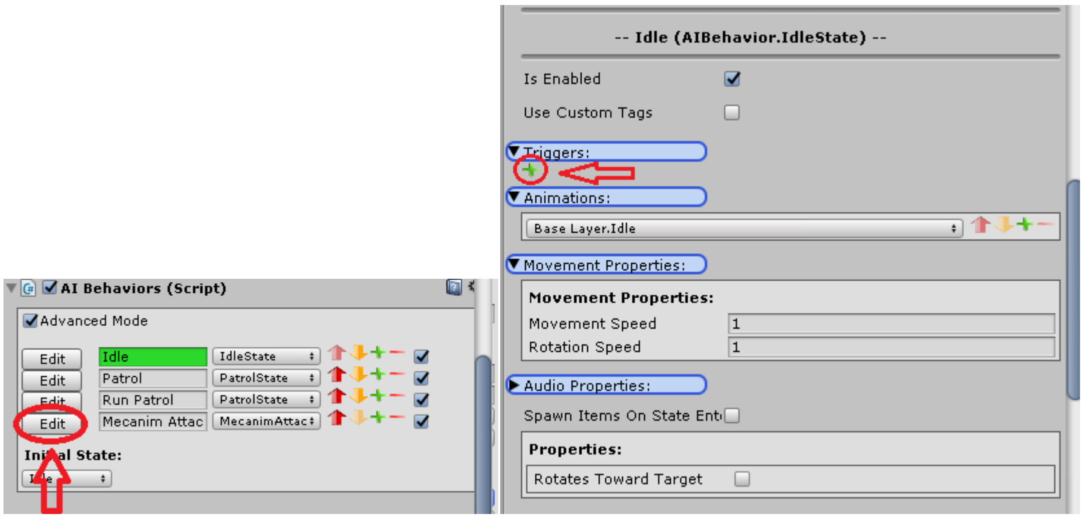
To add a global trigger just click the "+" button, choose the kind of trigger you want and set his properties.



AI Target Tags
Add
Player | Remove
Cache Point: Awake
Global Triggers:
Animation Component Callback:
Component: AIBehavior.MecanimAnimation
Method: OnAnimationState

3.2 State triggers

The state triggers are conditions that are checked inside a specific state. To add a trigger to a state click on the edit button left of the state in the "AIBehaviors" script in the inspector, and scroll down to the triggers section. Then click the "+" button to add a trigger.



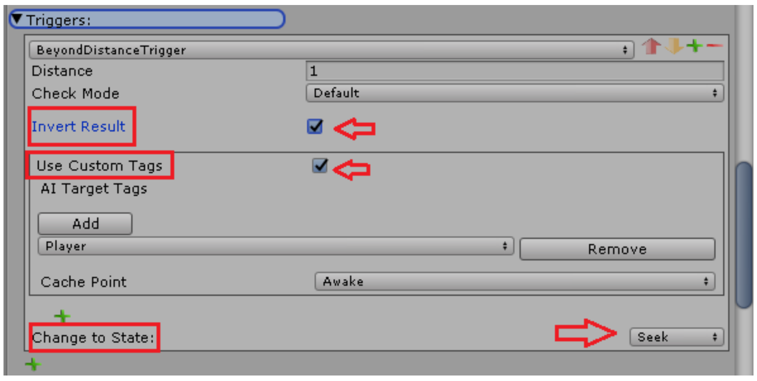
AI Behaviors (Script)
Advanced Mode
Edit | Idle | IdleState
Edit | Patrol | PatrolState
Edit | Run Patrol | PatrolState
Edit | Mecanim Attac | MecanimAttac
Initial State: Idle

-- Idle (AIBehavior.IdleState) --
Is Enabled: ✓
Use Custom Tags: ☐
Triggers:
Animations:
Base Layer.Idle
Movement Properties:
Movement Properties:
Movement Speed: 1
Rotation Speed: 1
Audio Properties:
Spawn Items On State Ent: ☐
Properties:
Rotates Toward Target: ☐

3.3 General Trigger Settings

There are some settings that are common to all the triggers, but apply to each trigger individually.

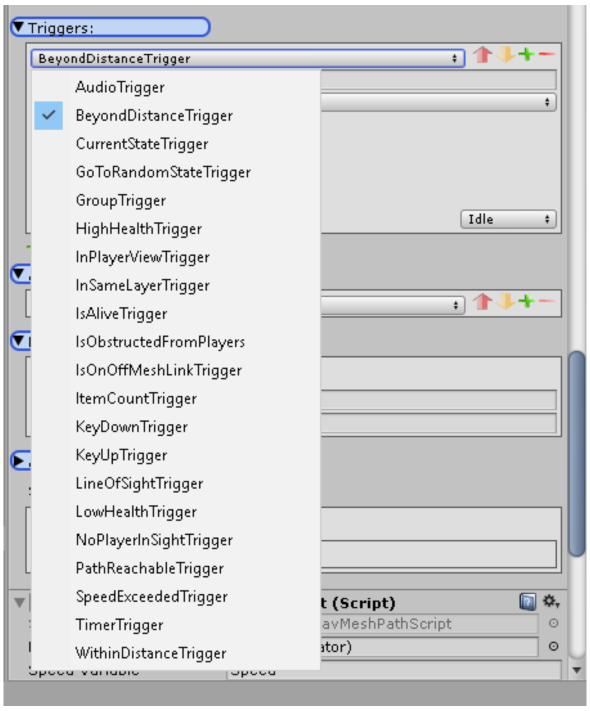
- **Change to State:** Is the state the AI will change into when the condition is met.
- **Use Custom Tags:** If checked you can set specific tag/s to the trigger.
- **Invert Result:** If checked will change the result of the trigger to the opposite. It will change the state when the condition is not met.



Triggers:
BeyondDistanceTrigger
Distance 1
Check Mode Default
Invert Result ✓
Use Custom Tags ✓
AI Target Tags
Add
Player Remove
Cache Point Awake
Change to State: Seek

3.4 Types of triggers

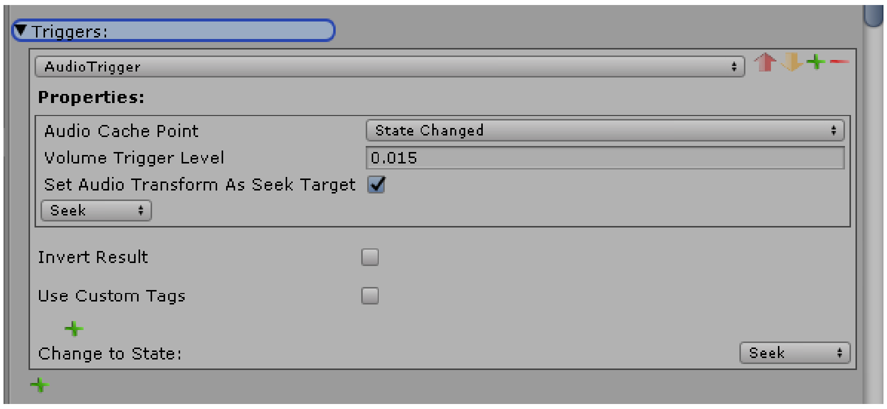
There are many different kinds of triggers, each one has different properties and settings. You can also write your own trigger with specific conditions, just extend the "BaseTrigger" script. It will automatically appear in the list of triggers.



Triggers:
BeyondDistanceTrigger
AudioTrigger
✓ BeyondDistanceTrigger
CurrentStateTrigger
GoToRandomStateTrigger
GroupTrigger
HighHealthTrigger
InPlayerViewTrigger
InSameLayerTrigger
IsAliveTrigger
IsObstructedFromPlayers
IsOnOffMeshLinkTrigger
ItemCountTrigger
KeyDownTrigger
KeyUpTrigger
LineOfSightTrigger
LowHealthTrigger
NoPlayerInSightTrigger
PathReachableTrigger
SpeedExceededTrigger
TimerTrigger
WithinDistanceTrigger

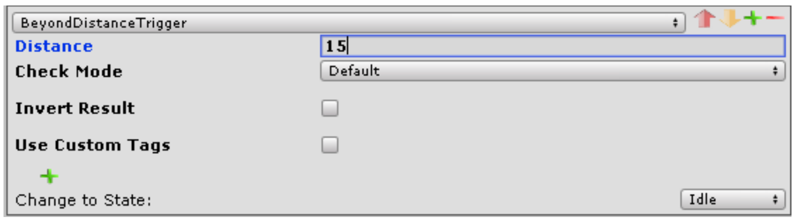
3.4.1 Audio trigger

Checks if the AI "heard" a sound from a GameObject with the specified tag. You'll need to figure out the "**Volume Trigger Level**" according to the distance you want him to hear (the higher this value is, the less he will hear). The "**Audio Cache Point**" specifies when the AudioSources on the scene are scanned. The "**Set Audio Transform As Seek Target**" option is to pick the AudioSource GameObject and set it as the target of a specific SeekState (in this case you should pick the same SeekState in "Change to State").


Triggers:
AudioTrigger
Properties:
Audio Cache Point: State Changed
Volume Trigger Level: 0.015
Set Audio Transform As Seek Target: ✓
Seek
Invert Result: ☐
Use Custom Tags: ☐
Change to State: Seek

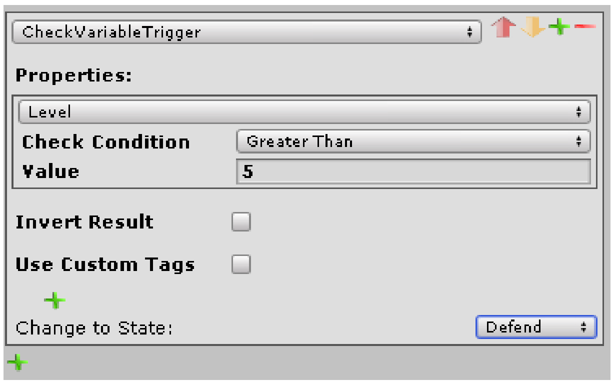
3.4.2 Beyond distance trigger

Checks if a GameObject with the AI target tag is farther than the specified "**Distance**". The "**Check Mode**" specifies if Any GameObject with the tag or All of them are beyond the distance.


BeyondDistanceTrigger
Distance: 15
Check Mode: Default
Invert Result: ☐
Use Custom Tags: ☐
Change to State: Idle

3.4.3 Check Variable Trigger

Uses the AI custom variables (Specified on Point 1.2), depending on the type of the variable you can make different checkings: For ints and floats you can check "Greater Than", "Less Than" or "Equal To", for booleans you can only check for "Equal To" as the other make no sense.


CheckVariableTrigger

Properties:

Level

Check Condition: Greater Than

Value: 5

Invert Result

Use Custom Tags

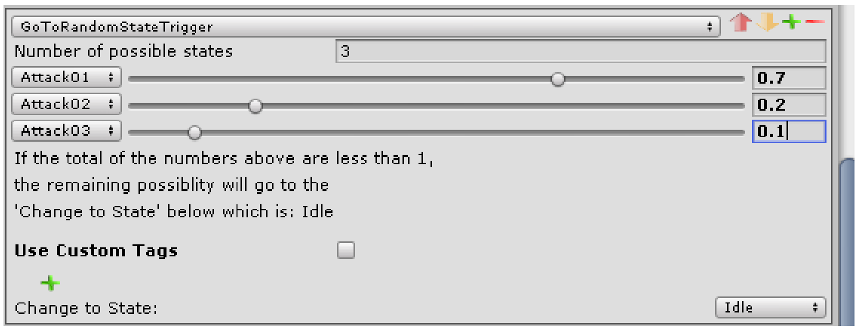
Change to State: Defend

3.4.4 Current state trigger

Checks if the AI is in a specific state, it has only sense to use it as a global trigger.

3.4.5 Go to random state trigger

Goes randomly to one of the possible states, the number on the right is the probability of each state. The combined amount of this numbers should be 1 or less.


GoToRandomStateTrigger

Number of possible states: 3

Attack01: 0.7

Attack02: 0.2

Attack03: 0.1

If the total of the numbers above are less than 1, the remaining possiblity will go to the 'Change to State' below which is: Idle

Use Custom Tags

Change to State: Idle

3.4.6 High health trigger

Checks if the health is over the specified amount.

3.4.7 In player view trigger

Checks if the AI is in the camera view. The "**Update View Bounds**" option sets when the bounds of the AI should be calculated (taking in account all the renderers of the children GameObjects). If the character doesn't change just leave in "Awake".

3.4.8 In same layer trigger

Checks if the AI is in the same layer as the GameObject/s with the specified tag/s.

3.4.9 Is alive trigger

Checks the health of the AI to see if it's alive. Generally used as global trigger.

3.4.10 Is obstructed from players

Checks if there is something between the AI and the player/target, no matter where its facing. You can set the maximum distance it will check.

3.4.11 Is on OffMeshLink trigger

Checks if the AI is on an OffMeshLink, (a connection point in the NavMesh where you can jump or fall down).

3.4.12 Item count trigger

Counts the GameObjects with the specified tag and compare it with the "Item Count" parameter according with the "**Comparison Condition**"

3.4.13 Key down trigger

Checks if a key was pressed. When the key is pressed it will change to the specified state.

3.4.14 Key up trigger

Checks if you stopped to press a key.

3.4.15 Line of sight trigger

Checks if a GameObjects with the specified tag is in the view of the AI (you have to set the view settings in the general AI properties). You can set a specific sight transform just for this trigger.

3.4.16 Low health trigger

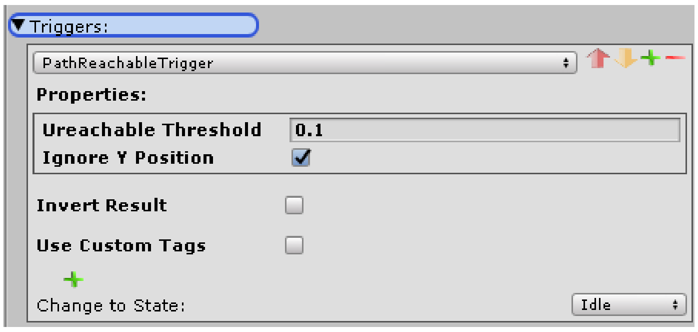
Checks the health of the AI if it is under the "**Below Health**" value.

3.4.17 No player in sight trigger

Is the opposite of LineOfSightTrigger, checks if there aren't any of the tagged objects in the view of the AI. The "**Trigger After Time**" value is how often the AI will do the checking.

3.4.18 Path reachable trigger

Checks if the AI can get to the target (the GameObject with the specified tag). It needs a NavMeshAgent. The "**Unreachable Threshold**" setting is the distance between the closest point the AI can get and the target. Check the "**Ignore Y position**" option to only check X and Z.

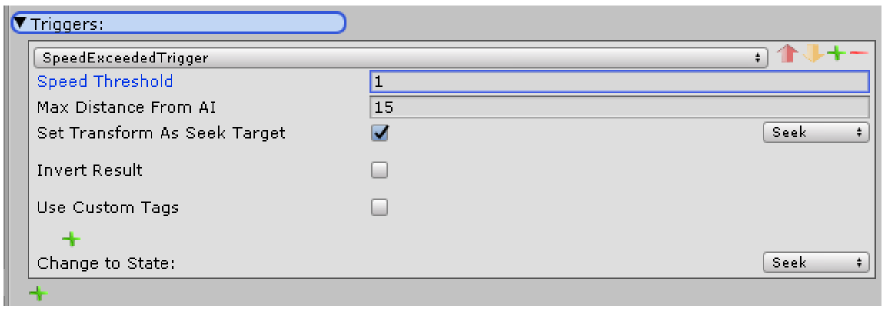


Triggers:
PathReachableTrigger
Properties:

Ureachable Threshold	0.1
Ignore Y Position	✓
Invert Result	☐
Use Custom Tags	☐
Change to State:	Idle

3.4.19 Speed exceeded trigger

Checks if an object with the specified tag moved faster than the "**Speed Threshold**" (m/s) within the "**Max Distance From AI**" distance. The "**Set Transform As Seek Target**" option is to pick the GameObject that moved and set it as the target of a specific SeekState (in this case you should pick that SeekState to be switched to, when triggered).



Triggers:
SpeedExceededTrigger

Speed Threshold	1	
Max Distance From AI	15	
Set Transform As Seek Target	✓	Seek
Invert Result	☐	
Use Custom Tags	☐	
Change to State:		Seek

3.4.20 Timer trigger

Goes to the specified state when the time is over. You can add some randomness to the time with the "**Plus Or Minus Duration**" value. The timer will be reseted according to the "**Timer Reset Mode**" (the most common is to use "On State Enter").

3.4.21 Within distance trigger

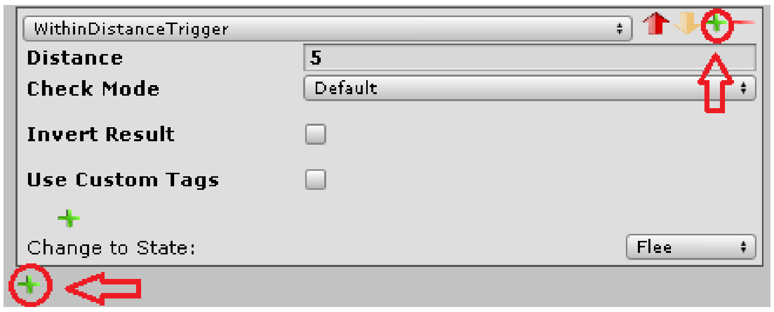
Checks if a GameObject with the specified tag/s is closest than the specified distance. The "**Check Mode**" specifies if Any GameObject with the tag or All of them are within the distance.

3.5 Nesting triggers

It is possible to combine triggers, this can be done in two ways:

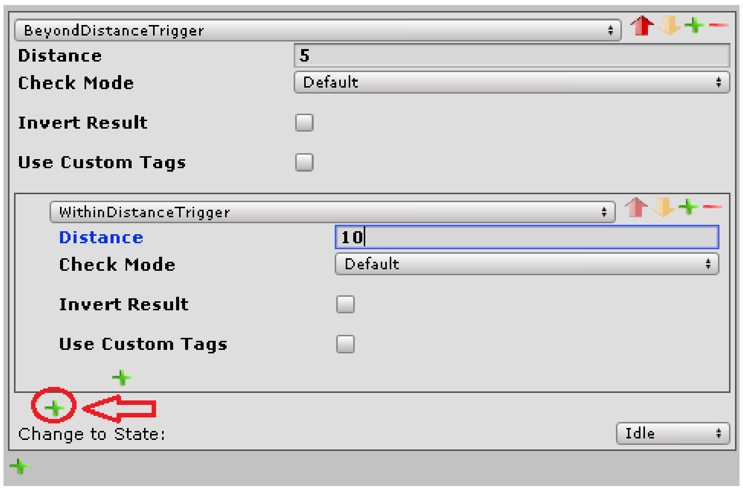
- putting two or more triggers near each other
- putting one or more triggers inside another trigger.

Putting triggers near each other make them act separately, each one will have his own state to change to when the condition is matched and the first one that is triggered will be the one that changes the state of the AI. For doing that click the green "+" button below a trigger (outside of it) or the one on the right.



WithinDistanceTrigger
Distance 5
Check Mode Default
Invert Result
Use Custom Tags
Change to State: Flee

Putting triggers inside other triggers is used when you want more than one condition for changing to a specific state. For doing that just click the green "+" button at the bottom of a trigger, this will create a new trigger inside it (it's like combining conditions in a if statement with "&"). When all the conditions are matched then it will change into the specified state.



BeyondDistanceTrigger
Distance 5
Check Mode Default
Invert Result
Use Custom Tags
WithinDistanceTrigger
Distance 10
Check Mode Default
Invert Result
Use Custom Tags
Change to State: Idle

You can make different combination of this to ways of combining triggers to make the best behavior for your AI.