

DOCUMENTATION 1.0

Work with PlayMaker 1.8 and Above, if you need that work with other Playmaker Version please contact me.

In the prefabs folder there are some implementations examples but if you need more specifics implementations or if you have any feature requests, please don't hesitate to contact me.

<mailto:oliver@axlplay.com>

On the website we always extend the documentation also the road- map of our

work. <https://axlplay.com>

PLAYMAKER ACTIONS LIST:

- FLEE:

Flee will move the agent away from the target with pathfinding. fledDistance The agent has fled when the magnitude is greater than this value.

lookAheadDistance

The distance to look ahead when fleeing.

target

The GameObject that the agent is fleeing from.

- PURSUE:

Pursue predicts where the target is going to be in the future. This allows the agent to arrive at the target earlier than it would have with the Set Destination Action.

targetDistPrediction

How far to predict the distance ahead of the target. Lower values indicate less distance should be predicated

targetDistPredictionMult

Multiplier for predicting the look ahead distance

target

The GameObject that the agent is pursuing.

-EVADE:

Evade is similar to the Flee except the Evade predicts where the target is going to be in the future. This allows the agent to flee from the target earlier than it would have with the Flee.

evadeDistance

The agent has evaded when the magnitude is greater than this value.

lookAheadDistance

The distance to look ahead when evading.

targetDistPrediction

How far to predict the distance ahead of the target. Lower values indicate less distance should be predicated.

targetDistPredictionMult

Multiplier for predicting the look ahead distance.

target

The GameObject that the agent is evading.

-PATROL:

Patrol moves from waypoint to waypoint.

randomPatrol

Should the agent patrol the waypoints randomly?

waypointPauseDuration

The length of time that the agent should pause when arriving at a waypoint

waypoints

The waypoints to move to.

-CAN SEE OBJECT:

Can See Object returns the object when it sees an object in front of the current agent.

usePhysics2D

Should the 2D version be used?

targetObject

The object that we are searching for. If this value is null then the objectLayerMask will be used

objectLayerMask

The LayerMask of the objects that we are searching for

ignoreLayerMask

The LayerMask of the objects to ignore when performing the line of sight check

fieldOfViewAngle

The field of view angle of the agent (in degrees)

viewDistance

The distance that the agent can see

offset

The offset relative to the pivot position

targetOffset

The target offset relative to the pivot position

returnedObject

The object that is within sight

-CAN HEAR OBJECT:

The Can Hear Object return the object when it hears another object.

usePhysics2D

Should the 2D version be used?

targetObject

The object that we are searching for. If this value is null then the objectLayerMask will be used

objectLayerMask

The LayerMask of the objects that we are searching for

hearingRadius

How far away the unit can hear

audibilityThreshold

The further away a sound source is the less likely the agent will be able to hear it. Set a threshold for the the minimum audibility level that the agent can hear

offset

The offset relative to the pivot position

returnedObject

The returned object that is heard

-WANDER:

Wander moves the agent randomly throughout the map.

wanderDistance

How far ahead of the current position to look ahead for a wander

wanderRate

The amount that the agent rotates direction.

-PURSUE:

Pursue predicts where the target is going to be in the future. This allows the agent to arrive at the target earlier than it would have with the Set Destination Action.

targetDistPrediction

How far to predict the distance ahead of the target. Lower values indicate less distance should be predicated

targetDistPredictionMult

Multiplier for predicting the look ahead distance

target

The GameObject that the agent is pursuing.

-SEARCH:

Search will search the map by wandering until it finds the target. It can find the target by seeing or hearing the target.

wanderDistance

How far ahead of the current position to look ahead for a wander

wanderRate

The amount that the agent rotates direction

fieldOfViewAngle

The field of view angle of the agent (in degrees)

viewDistance

The distance that the agent can see

ignoreLayerMask

The LayerMask of the objects to ignore when performing the line of sight check

senseAudio

Should the search end if audio was heard?

hearingRadius

How far away the unit can hear

offset

The offset relative to the pivot position

targetOffset

The target offset relative to the pivot position

objectLayerMask

The LayerMask of the objects that we are searching for

linearAudibilityThreshold

The further away a sound source is the less likely the agent will be able to hear it. Set a threshold for the the minimum audibility level that the agent can hear

returnedObject

The object that is within sight.

-FLOCK:

Flock moves a group of objects together in a pattern.

agents

All of the agents that should be flocking

speed

The speed of the agent

angularSpeed

Angular speed of the agent

neighborDistance

Agents less than this distance apart are neighbors

lookAheadDistance

How far the agent should look ahead when determine its pathfinding destination

alignmentWeight

The greater the alignmentWeight is the more likely it is that the agents will be facing the same direction

cohesionWeight

The greater the cohesionWeight is the more likely it is that the agents will be moving towards a common position

separationWeight

The greater the separationWeight is the more likely it is that the agents will be separated

-SHOOT

The scene ShooterDemo is an example of this action. This action shoots a gun.

timeBetweenBullets

The time that the agent waits between each shoot **range**

The distance of the bullet.

ShootableMask

The layer of the gameobjects that the agent can shoot

GunGameObject

The gameobject where the bullet comes out

PlayShootAnimation

Would you like to play an animation when the agent shoot?

TriggerShootAnimation

The animation that will be played when the agent shoot. If PlayShootAnimation is false you haven't to fill this

OnPlayerHit

The event that will be fire when the agent hit the target

AgentCanReload

Do you want to the agent can reload?

AmmunitionForCartridge

A number of bullets that the agent can shoot before of reload. If AgentCanReload is false don't fill this field

TriggerAnimationReload

The parameter in the animator.If AgentCanReload is false do not fill this field

GunLight

Do you want to turn on a light when the agent shoot?.

PlayAudioOnShoot

Would you like to play an audio when the agent shoot?.

SecondsReloading

The seconds that the agent will take reloading

PlayMuzzleFlash

Do you want that the pistol plays a muzzle flash when the agent shoot?.

Audio

The audio that will be played when the agent shoot. If PlayAudioOnShoot is false don't fill this field.

HitPoint

The hit point of the shoot(Only if hit a ShootableMask) .

-TARGET REACHABLE

Field Of View returns an object when it sees an object within of the field of view, without an object blocking the target.

viewDistance

Is the distance that the agent can see.

viewAngle

Is the field of view of the agent.

targetMask

Is the layer of gameobjects that you want to detect.

obstacleMask

Is the layer of obstacles that can block the sight between agent and target. Example: A wall.

numRaysScene

Is the number of rays that you can see in scene view (Gizmos).

StoreTarget

Store the gameobject seen closest

hitEvent

Event to send if get a target object

-AGENT STOP

This action stop the NavMesh agent

-AGENT RESUME

This action resume the NavMesh agent

-CAN SEE FROM SIGHT

Can see from sight.

goFromSight

The object From sight.

usePhysics2D

The field of view angle of the agent (in degrees).

fieldOfViewAngle

The distance that the agent can see.

viewDistance

The distance that the agent can see.

offset

The offset relative to the pivot position.

returnedObject

The object that is within sight

finishEvent

This event will fire when the target was seen

targetObject

The object that we are searching for. If this value is null then the objectLayermask will be used

HitLayerMask

The LayerMask of the objects that we are searching for.

IgnoreLayerMask

The LayerMask of the objects to ignore when performing the line of sight check

targetOffset

The target offset relative to the pivot position

angleOffset2D

The angle offset relative to the pivot position 2D

UFPS INTEGRATION

UFPS integration is extremely simple:

- 1.- Go to Assets/Easy AI For Playmaker/Add-On/UFPS and import UFPS.unitypackage.
- 2.- Search for all the UFPS bullet and explosion prefabs in the project view and set RequireDamageHandler to Unity Message.
- 3.- Ensure that the character that will receive the damage of the UFPS Character contains the script [TakeDamage.cs](#)
- 4.- The action "Make Damage" will apply damage to the UFPS Character, you could use this action with "Shoot" and fire it the event "Playerhit" (Inside the action "Shoot")

-TAKE DAMAGE (SCRIPT C#)

Get the damage infringed for the UFPS Character

Name Of Variable That Receive The Damage

The name of the playmaker variable type float where it will assign the damage received.

Name Of Fsm

The name of the FSM where is the variable "**Name Of**

Variable That Receive The Damage".

Name Of Event That Run On Receive Damage

The name of the event that fire up when our character receive damage (infringed for an UFPS Character).

ACTION PLAYMAKER: -MAKE DAMAGE

Infringe damage to the UFPS Character

Target

The UFPS character to infringe damage

Damage

The amount of damage to infringe

Thanks you for your purchased.

If you have any feature requests or need support please let us know.

Contact us: oliver@axlplay.com