Advanced AI Pro 6.0

Documentation

By ABBAS

2014

Contents

i. Introduction:

Thank you for purchasing Advanced AI Pro system for Unity 3D engine, I developed this plugin to be similar to most industry standard AI systems which we see in nowadays games, designed for enemy behaviors, thus NPC and friendly behaviors.

This AI system could cover any type of enemy behaviors, including melee or ranged attacker, with the ability to wander in a user-specified area to search for the target, as well as, NPC, and friendly behaviors.

AI has ears to hear any noise made by the target within a specified region, also all AI's chasing and attacking behaviors are based on its field of view.

All parameters could be manipulated easily in the inspector to satisfy your needs, no other coding/scripts are required as we will see soon.

ii. Before you start:

As Advanced AI is based on Unity built-in navigation API, so before adding an AI you need to bake your scene for navigation. It is very simple; well when your level or example scene is ready to test do the following:

1- Flag all static game objects in the scene with the « Static » flag which you can find in the top-right of the inspector. Static means that this game object doesn't move so it will be considered as static when navigation's calculation, don't forget to flag the « floor » too as static, also (walls, and other geometry).

2- Open « Navigation Window « from Unity menu at the top, then click « Bake » button, this can take some time depending of the size of your scene, when it is done you should be able to see that walkable area in the scene has become blue. Remember that you can select certain game objects from the scene then in Navigation window you choose « Not Walkable » so this area can't be accessed by the AI.

3- **Important**: You need to create 2 new unique layers, one is for the target game object, and the other one is for the AI's view obstruction game objects (like walls). Create them, and then assign them to your target, walls, and floor…..game objects.

4- At last, your character game object must have an "Animation" component attached with all your character's animation clips listed in it.

iii. Adding Advanced AI:

1- First be sure that the game object of your character has an "Animation" component added, with your animation clips list.
2- Select the game object of your character from the scene hierarchy.
3- When the game object is selected, go to Unity main menu: "***GameObject > Advanced AI Pro"***, then add Advanced AI.
4- Done, your character now is setup, **no need** to add additional scripts or components to the newly created AI game object.

You will notice that a "Capsule Collider" component is added to your newly created AI game object, this is important so that the AI can have physical collision with the world, as well as taking hit from your target (player) attacks. Change its values (height, center, and radius) to make it fit your character model.

Note: The "Projectile" component which you can add to your projectile game object can recognize that capsule collider attached to the AI so it can cause damage to that AI.

Note: If your original character game object has a collider on it then you have to delete it.

There are five main classes of Advanced AI:

- The first is (**Enemy**) which has attacking behavior (melee or ranged or both of them).
- The second is (**NPC Aggressive**): It starts as a neutral non-attacking NPC, but when you attack it, it will transform into an enemy behavior.
- The third is (**NPC Passive**): This has a neutral passive behavior, it can wander around, but when you attack it, or when you are within its view range (or both cases) it will try to flee away from you, and then continues to wander/idle (This class can be implemented for animal behaviors by example, or city civilians).
- The fourth is (**Companion**): This is a friendly AI which can be your companion, it follows your commands to follow you or to stop in place, and it can be attacked by enemies when seen.
- The fifth is **(Defender Ally)**: This is your ally AI which can be your defender and companion. It follows your commands to follow you, or to stay in place and guard its area, and it will attack your enemies when seen, it has melee and ranged attack modes.

  - **Note**: You don't need to setup the animations' wrap mode for your character; all needed animations are setup automatically in Advanced AI module.

  - **Note:** The animation type of your imported character model must be set to «Legacy» in the rig tab (if you are using Unity 4).

All you need to do now to setup your AI is to play with the exposed parameters from the inspector; you will find that your character became a child of the AI parent game object, with a new « _ProjectileOrigin » game object added too.

You can change the name of main AI game object, thus your character name, but **Do Not** change the name of "_ProjectileOrigin" game object.

If your AI does ranged attack, then you must set the position of **"_ProjectileOrigin"** child game object to match your projectile launching point.

iv. Setup:

- AI inspector Parameters (shared between Enemy, and NPC classes):

Note: The parameters are in alphabetic order, so you can easily search for the parameter which you want to know about;

- **Acceleration:**

Maximum acceleration for the AI (float).

- **Alert Distance:**

This is the alert range for the AI (see "Can Alert" parameter for more information).

- **Alert Sfx:**

The sound effect to be played after the AI gets hit to alert nearby enemies (or NPC Aggressives) to ask for help.

- **Arrival Distance:**

This float value is the distance between the AI and the target which lets us to consider that the AI has arrived to his target.

- **Attack Mode:**

- **Melee Attack:** The AI will do a melee attack, if the target is within the view field of the AI then the AI will start to chase the target and then attack it when it arrives to the "Arrival Distance" of the AI.

- **Ranged Attack:** The AI will do a ranged attack whenever the target is within its "Shoot Distance" range.

- **Melee_Ranged:** The AI will use both melee and ranged attacks depending on its distance from the target, if the target's distance is equal or inferior to the AI's "Arrival Distance" then the AI will do melee attack, otherwise the AI will do ranged.

- **Blood Decal Dead:**

If assigned then the AI will leave that blood decal prefab on the ground when it dies.

- **Can Alert:**

If checked, then AI enemy, will alert other nearby enemies, within its “Alert Distance” range, when it gets hit, so they will come to the injured AI trying to give help by going and searching the area for the target.

- **Can Retreat:**

If checked, then the AI retreat from the battle and run away from the target when its health points are equal or low to “Retreat Health” value.

- **Can Hear:**

Check this to let AI hear sounds played by the target.
Note: The target game object, or any of its child game objects, must have an “Audio Source” component attached to it, the AI will recognize any sounds played by the target within the hearing radius of the AI.

- **Chasing Anim:**

Chasing animation clip of the AI character.

- **Chasing Sfx:**

The Sfx audio clip to be played when the AI chases its target (in melee attack mode).

- **Chasing Speed:**

The speed of the AI when chasing the target (float).

- **Second Targets Layer:**

Assign to this layer any gameobject which you want your AI enemy to attack, it can be Companion AI, defender ally AI, or any other gameobject. The second target gameobject should have a collider component attached to it. And you can assign as many gameobjects to this layer, contrary to “Main Target Layer” which should be unique and assigned only to your player character.

- **Damage Amount:**

The damage amount (float) which the target will receive when it is attacked by the AI, set the size of Damage value according to your Melee Attack Anim’s size, for example if you have two melee attack anims then the first damage value well be assigned for the first melee animation, and the second for the second one……ect.

- **Damage Method Name:**

This is the name of the method (function) that will be sent to the target (player) with the damage amount when the target gets hit (from melee or ranged projectile).

- **Damage Delay:**

This is the amount of time (in seconds) to send the damage message to the target after playing the melee animation clip. Set this value so the damage delay is synchronized with you melee animation clip.

- **Death Anim:**

Animation clip to be played when AI's health reaches zero or less.

- **Decal Destroy Delay:**

Time in seconds before removing projectile's decals game objects from the scene (Default is 10 seconds).

Note: Having a long destruction time for decals, and having shot many projectiles (in action-packed games by example) may slow up your game, so be sure that this time value is not so long.

- **Detection Sfx:**

The sound clip to be played when the AI discovers (see) its target, only possible for melee attack.

- **Die Sfx:**

Sound clip to be played when the AI dies.

- **Disappear On Death:**

If checked, then the AI game object (with its children) will be destroyed after finishing playing the death animation and death Sfx sound.

- **Disappear Delay:**

The delay in seconds to destroy the AI game object after finishing playing the death animation and death Sfx sound.

- **Force Melee Conclude:**

If checked then the AI will conclude its melee animation before chasing the target (when this latter gets away from the AI).

- **Go Idle Radius:**

This float value is the radius beyond it the AI will go to idle mode, for example when the target is very far from the AI (represented by white sphere gizmo in the editor window).

- **Got Hit Anim:**

The animation clip of the clip to be played when the AI got hit.

- **Got Hit Sfx:**

Sound clip to be played when the AI gets hit.

- **Health Points:**

The health amount of the AI (float), when it reaches 0 or a negative value, the AI will die.

- **Hearing Distance:**

Within this distance from the target, the AI will be able to hear any sounds playing by Audio Source of the target (represented by a blue sphere gizmo in the editor window).

- **Hear Volume Min:**

This is the minimum audio volume emitted by the target which can be heard by the AI, can be useful for stealth games.

- **Hit Particle Fx:**

If assigned, then it will be instantiated when the AI gets hit (e.g. bloot splats).

- **Idle Anim:**

The animation clip to be played when the AI idles.

- **Idle Sfx:**

The Sfx clip to be played when the AI idles.

- **Intelligence Mode:**

- **Stupid**

- **Smart:** The AI is clever; it will run to the last known position of the target when the latter is out of AI's vision range. Also the AI will run towards the shooting point when it gets hit, from the target, from far or from behind.

- **Drop Item On Death:**

If assigned then the AI will drop this game object (prefab) when it dies, this can be useful if you want the AI to leave ammo, key, pickups…ect.

- **Drop Item Offset:**

The offset position of the instantiated game object to the position of the AI.

- **Look At Speed:**

Float value of the speed when the AI rotates to look towards the target, greater the value faster the AI will detect the player when it gets hit, or heard.

- **Melee Attack Anim:**

Choose how many animations you want your AI to randomly play when attacking the target, and then write the name of each animation clip. (The AI will randomly choose between those melee attacks anims after each interval).

- **Min (and Max) Idle Interval:**

These two parameters determine the minimum and maximum time in seconds for which AI blend into the idle state when wandering or patrolling waypoints. So the idle interval time is randomized between those two values.

- **Melee Attack Sfx:**

The Sfx clip to be played when the AI attacks, set the size of Sfx clips according to your Melee Attack Anim's size.

- **Attack Interval (for Melee):**

The interval time (in seconds) between each melee attack.

- **Patrol Mode:**

  - **In Place**: The AI will idle in place only.

  - **Dynamic Wandering:** The AI will randomly wander within a specified area (determined by "Wandering Radius" parameter) to search for the target if the latter is not within the view field of the AI.

  - **Waypoints:** The AI will follow the waypoints added by "Waypoint Editor", it will follow them in order (0 > 1 > 2 >….) and when it reaches the last one it will repeat the cycle from the start.

- **Proj Impact Fx Dest Time:**

Time (in seconds) before destroying the FX particle system played on projectile's impact point.

- **Projectile Decal:**

This is the texture asset of the decal you want to use for AI's projectile.
Note: Make sure that the texture image has transparency (alpha channel) around the "Bullet hole" so the surrounding information will be ignored.

- **Projectile Destroy Delay:**

Time (in seconds) before destroying the projectile game object.

- **Projectile Impact Fx:**

The particle effect prefab to be played on projectile's impact point on objects (like smoke particles on bullet holes).

- **Projectile Impact Sfx:**

The Audio clip to be played when the projectile hits an object (like bullet's sound).

- **Projectile Fx:**

The particle system to be played at the "_ProjectileOrigin" when the projectile is launched. It can be shuriken or legacy particle systems. For example if the ranged attack is using a firearm then "Projectile Fx" can be the muzzle flash of the weapon.

- **Projectile Delay:**

This value determines after which amount of time (in seconds) the projectile will be launched after playing the ranged animation. Set this value so that the projectile launching is synchronized with your ranged animation clip.

- **Projectile Prefab:**

The game object prefab of the projectile used for the AI ranged attack.

- **Projectile Velocity:**

The velocity of the projectile (float).

- **Ragdoll Play Die Anim:**

Choose if the AI character should or not play its death animation before transforming into a ragdoll (if Ragdollify is checked).

- **Ragdoll Prefab:**

The prefab of the ragdoll game object which you made for your AI character, assign it here.

- **Ragdollify On Death:**

If checked then the AI character will be transformed into the ragdoll game object you assigned earlier when it dies, and will play the death animation before staying as a ragdoll in the ground influenced by physics forces.

- **Ranged Attack Anim:**

The animation clip of the character to be played when the AI attacks the target from distance (like shoot animation).

- **Ranged Attack Sfx:**

The Sfx clip to be played when ranged attacking (gun shooting sound…..ect).

- **Shoot Interval:**

The interval time (in seconds) between each ranged attack.

- **Projectile Damage:**

The damage amount (float) which the target will receive when it is attacked by the AI's projectile.

- **Reload Anim:**

The animation clip to be played when reloading.

- **Reload Sfx:**

Reloading sound effect.

- **Reload Interval:**

This is the number of ranged attacks (shoots) before reloading.

- **Retreat Anim:**

  The animation clip to be played by the AI when retreating, if not assigned then the AI will play its chasing animation.

- **Retreat Sfx:**
  The sound effect to be played during the retreat, of not assigned then chasing sfx will be played instead.

- **Retreat Distance:**

This is the distance to which the AI will retreat.

- **Retreat Health:**

This is the health value for which the AI will retreat, when its health level is equal or below to this value.

- **Rotation Speed:**

The angular speed (float) of the AI when it is wandering or chasing the target.

- **Shoot Range:**

The maximum distance between the AI and the target in within the AI (ranged) can shoot at the target.

- **Target:**

The transform of the target (this is generally the player).

- **Target Unique Layer:**

The layer of the main target (the player character for instance). This layer must be unique, and not shared by other game objects other than your main target gameobject. If you want AI enemy to attack second targets then assign them to "Second Targets Layer".

- **View Obstruction Layer:**

Choose the layer of the obstacles that obstruct the view of the AI (like walls).

- **View Sphere Center:**

Set the center position of the view sphere of the AI, the position is relative to the transform point of the AI game object. By default the position is set so that the view sphere is just in front of the AI eyes (to forward direction).

- **View Sphere Radius:**

The radius of the view sphere of the AI (or target-detection sphere). The target will be seen/detected by the AI when it is within this view sphere radius (represented by a red sphere gizmo in the editor window).

- **Wandering Anim:**

The animation clip to be played when the AI when it is wandering.

- **Wandering radius:**

The radius of the area around the AI, within this area the AI will randomly wander (for "Dynamic Wandering" patrol mode). (Represented by green sphere gizmo in the editor window).

- **Wandering Sfx:**

The Sfx audio clip to be played when the AI wanders.

- **Wandering Speed:**

The speed of the AI when it is wandering.

- **Wounded Idle Anim:**
  The animation clip to be played when AI is idling after retreating. If not assigned then AI will play its normal idle animation.

- **Wounded Idle Sfx:**
  The sound effect to be played when idling after retreating. If not assigned then Idle Sfx will be played instead.

- Special Parameters (For NPC Passive only):

- **Flee Anim:**
  The animation clip to be played when you attack the NPC, so it will flee (run away).

- **Flee Distance:**
  This is the distance to which the AI will flee when it gets hit.

- **Flee Mode:**
  - **On_Sight:** The AI will flee only if the target is within its view range (represented by the red sphere gizmo in the editor window).

  - **On_Hit:** The AI will flee only if it gets hit from the target.

  - **Both:** The AI will flee on both cases above.

- **Flee Sfx:**
  The sfx audio clip to be played when the AI flees away from its target.

- **Flee Speed:**
  The speed (float) of the NPC when it flees.

- Special Parameters (For Companion AI only):

- **Command Range:**

The distance from the target in which the AI can hear orders, so you (the target) have to be within this range to be able to give commands to your companion AI, this range is represented by a green sphere gizmo in the editor window.

- **Follow Anim:**

The animation clip to be played when the AI follows you.

- **Follow Command Key:**

The keyboard key to be assigned for “follow” command.

- **Follow Sfx:**

The audio clip to be played when following (it can be footsteps by example).

- **Follow Speed:**

The speed (float) of the AI when it follows you.

- **Stop Command Key:**

The keyboard key to be assigned for “Stop” command

- Special Parameters (For Defender AI only):

- **Enemy Unique Layer:**

This is the layer of the enemy AI, so the defender can recognize your enemies in this layer. Be sure that this layer is unique and not used by other game objects other than enemies AI.

- **Command Range:**

The distance from the target in which the AI can hear orders, so you have to be within this range to be able to give commands to your Defender AI, this range is represented by a green sphere gizmo in the editor window.

- **Follow Command Key:**

The keyboard key to be assigned for the “follow” command.

- **Stop Command Key:**

The keyboard key to be assigned for the “Stop” command.

Note 1: Put all your enemies in the same layer e.g.; “Enemy”.

Note 2: You cannot give commands to your defender(s) ally when it is attacking enemies.

v. Waypoint Editor:

Waypoints patrol mode is available for enemy class, as well as, NPC aggressive AI. In this patrol mode the AI will follow a series of waypoints added by you in order, and then repeat the cycle when it arrives to the last waypoint in the group.

Contrary to “Dynamic Wandering” the waypoints added to the scene are static; this means that they don’t move with the AI, so it could be useful if you want the AI to patrol a specific area. “Dynamic wandering” is a random patrolling, so AI will choose its waypoints automatically within the wandering range which you can control from the inspector.

To be able to use the Waypoint Editor, first select “Waypoints” in “Patrol Mode” parameter in the AI inspector.

To add a group of waypoint, click **“Add Waypoint”** button, you will find it in the scene with “WP” icon, a new parent game object will be created in the scene hierarchy holding all waypoints game objects for the selected AI, and named as the “Waypoints Group” text field.

You can open this group and select a waypoint to position it exactly as you want it to be, and then click again on “Add Waypoint” to add a second waypoint….ect. If you want to delete a waypoint click on **“Delete Last”** which will delete waypoints from latest one to the one before……ect.

Important: Don’t duplicate/delete the waypoints game objects manually from the scene, use only the buttons in “Waypoint Editor” to add/delete. If you made a mistake, then you can delete the existing waypoints group and start a new group by clicking on “Delete All”.

You can change the name of waypoints group (parent) game object by changing the text field: “Waypoints Group” and then clicking on **“Change Name”** button.

The **“Show/Hide GUI**” button is used to show or to hide the “WP” icons and the position handles displayed on each waypoint game object.

Also in **“Lines color”** you can choose the color of the lines connecting the waypoints, so you can recognize waypoint groups belonging to different AIs in the scene.

vi. Others setup:

1- Projectile:

From ***“GameObject > Advanced AI Pro > Other”*** submenu, click ***“Add Projectile component”*** when selecting your projectile game object. This is a universal projectile behavior, it can do damage to the player, as well as all Advanced AI types (enemy, defender, companion, NPC).

Add this component only if you want to implement a projectile behavior for your player when shooting, so it can hurt other AI(s).

2- Target (player):

From ***“GameObject > Advanced AI Pro > Other”*** submenu, click ***“Add Player Health component”*** when selecting your player game object, you can set the health amount from the inspector; this script handles attack damage received from the enemy (melee and projectile).

Important: The target should have a collider component attached, so it can be recognized by the AI, any collider type works (including “Character Controller”).

Note: The damage method name used in "Player Health" component is "SubtractHealth", however if you are using your own player controller which doesn't use that script component then you can simply define the damage method name in AI Melee/Ranged Attack parameters group which will send damage amount (in float value) to the player.

Note: If you decide to store your AI game object as a prefab to be instantiated later (through a script) then you should tag your target game object as **"Player"**, as the "Target" slot cannot be assigned when the AI is as prefab.

3- Make AI receives damage:

To make the AI receive damage, it is not simpler than that, you just have to call this function on the AI game object:

SendMessage("GotHit", damage amount(float));

A capsule collider is added automatically to your character AI when adding any of the AI classes from "Advanced AI" Component menu, which has two functions:

- First as a physical collider for the AI with world/player.
- And second, as collision recognition for the projectile, so any projectile game object having the "Projectile" component attached colliding with any of the AI classes will call the previous function to hit the AI, by the damage amount determined in the "Projectile" class.

vii. Jumping AI:

Note that to bake "Off Mesh Links" is a **Pro feature** only of Unity, and it is not available for the free license.

If you have gaps in your scene and you want your AI to jump off those gaps when chasing the target or even when following you, then you need to bake Off Mesh Links data from the "Navigation" window:

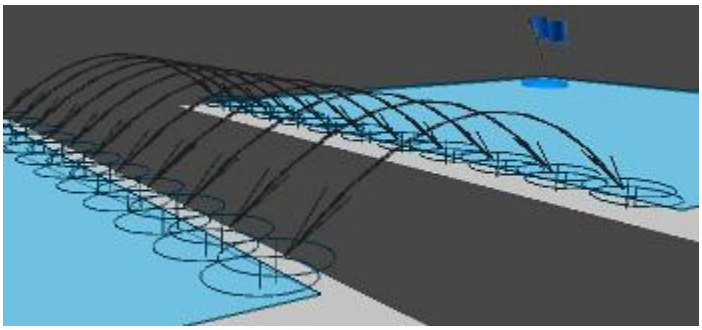
1- Choose your 2 disconnected static geometry from the scene.

2- In "Navigation" window under "Object" tab check "OffMeshLink Generation".

3- Under "Bake" tab, in "Generated Off Mesh Links" you have to change the "Jump Distance" from 0 to your desired positive value. (This is the maximum distance between the 2 surfaces for the AI to jump off).

4- Now you can bake your scene.

You will notice after baking the scene that OffMeshLinks data are visible now, like in the photo:



Now, in your AI inspector, under “Animations”, choose your jump animation, and optionally under “Sounds” a jump Sfx for it. Also in “Navigation” section you can set the jump speed of the AI.

AI will play its jump animation seamlessly depending on the animation length and the gap distance.