

WORQ MEDIA STUDIOS

**ADVANCED ENEMY AI
(AEAI)**

Thank you for purchasing this product. For questions regarding any of our products, and further inquiries, please contact us via the email worqmedia@gmail.com

HOW TO SETUP

Before proceeding, please ensure that you have baked a navigation mesh that the AI entity is going to walk/patrol on. This AI system depends on the default unity Navigation mesh.

Section A (Importing)

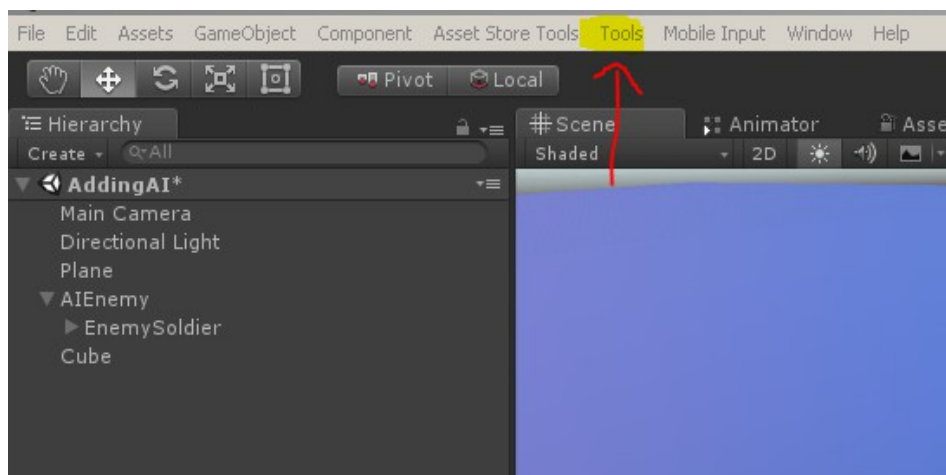
1. Import Advanced Enemy AI .unitypackage into your project.
2. Open the Advanced Enemy AI folder and locate the Editor folder
3. Cut the contents of the Editor folder and place them into your root Editor folder. (if you do not have a root editor folder, create a new folder in your main assets directory and name it Editor).
4. The AEAI sub-menu should now appear on your menu bar under “Tools” menu

Section B (Setting up AI)

There are 2 ways to setup an entity to work with this AI system. We begin with the automatic way which takes no more than a few clicks.

METHOD 1 (Automatic setup)

1. Select entity or entities that you wish to add AI to in your hierarchy.
2. Open the AEAI sub menu which can be located under tools menu on your menu bar.



3. Click on “Create AI” and viola! You have successfully added AI to the selected entity/entities

Upon doing so, you will see that the objects to which AI was added are enclosed in a new GameObject named “AIEnemy”. This is now the reference to the AI and defines the shell for that entity. You may move or rename this AI entity shell, but please keep the structure of child elements as this AI system depends of it.

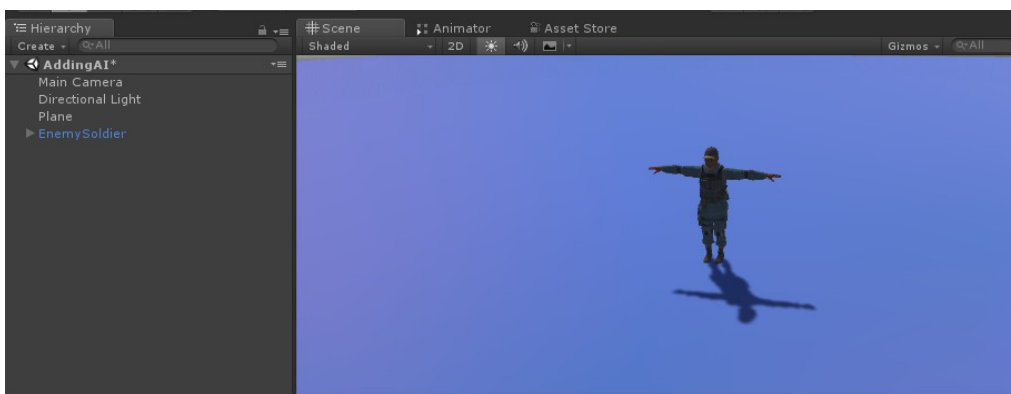
Adding patrol way points for AI entity:

To setup way points for a particular AI entity, start by selecting the main entity shell (titled “AIEnemy”, if name is not changed). Next go to the AEAI sub-menu that is located on your project's menu bar under tools. Now select “Add New Waypoint”. A new patrol waypoint has now been added to your AI entity's patrol route and is ready to use. This may be done as many times as needed as long as the AI entity shell is selected. **NB:** waypoints can only be added to AI entities and an exception will be thrown if you try to add to any GameObject that has no AI setup.

METHOD 2 (Manual Setup)

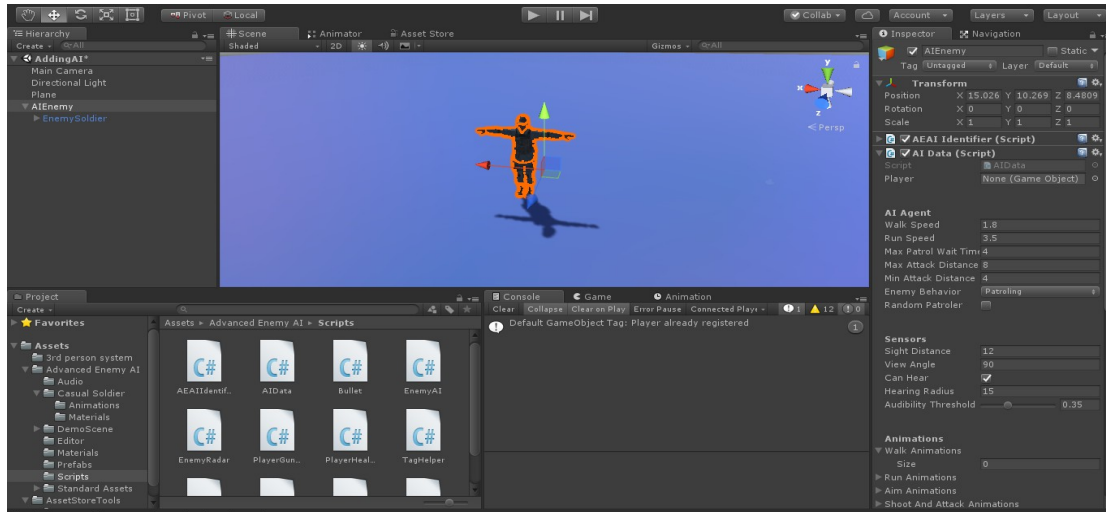
This is the manual setup method and should be followed exactly as specified here. We will use the term AIEnemy to refer to the main shell of the AI entity. We will use the term EnemySoldier to refer to the entity to which we want to add AI.

1. Begin by dragging the gameObject to which you need to add AI onto your hierarchy. This may be your enemy fbx model for instance.

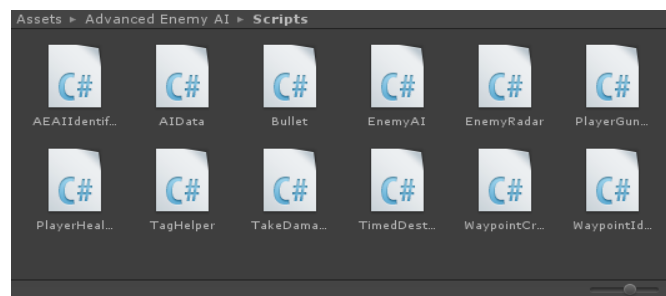


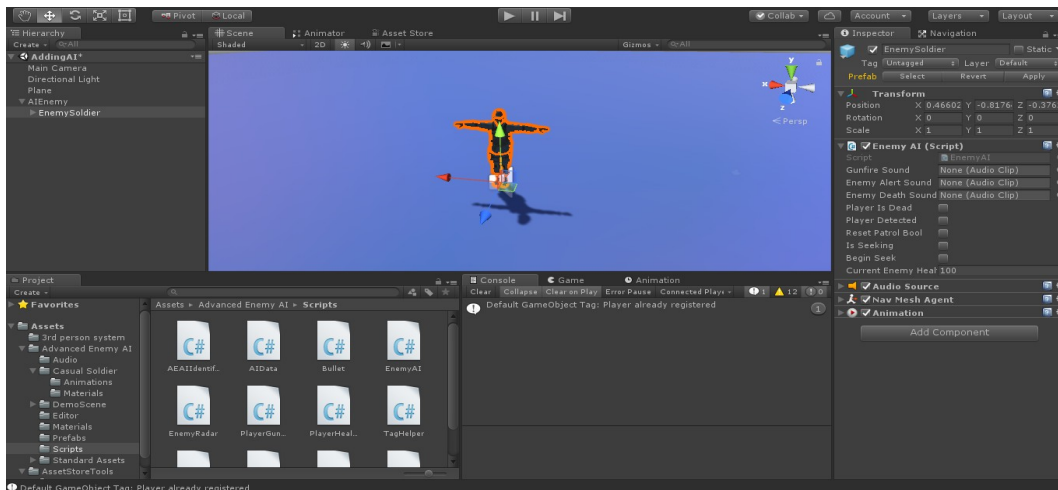
2. Next, create a new empty GameObject in your hierarchy. This is going to be the main AI entity's shell, so rename it to “AIEnemy” or something similar. We will use the name “AIEnemy” for the purposes of this example.

3. Locate the script titled “AEAIIdentifier” (located in the Main AEAI folder, under Scripts sub-folder) and add to the AIEnemy (the empty game object we created in step 1 above)
4. Locate the script titled “AIData” (located in the Main AEAI folder, under Scripts sub-folder) and add to the AIEnemy (the empty game object we created in step 1 above)



5. Next, find the script titled “EnemyAI” (located in the Main AEAI folder, under Scripts sub-folder) and add to the entity to which you want to add AI to (the EnemySoldier in this example)





6. Finally, parent the EnemySoldier GameObject to the AIEnemy GameObject by simply dragging the EnemySoldier and dropping onto the AIEnemy.

Your EnemySoldier entity now has AI added and is ready to use, but lets also create way points so the Enemy can patrol.

Adding patrol way points for AI entity:

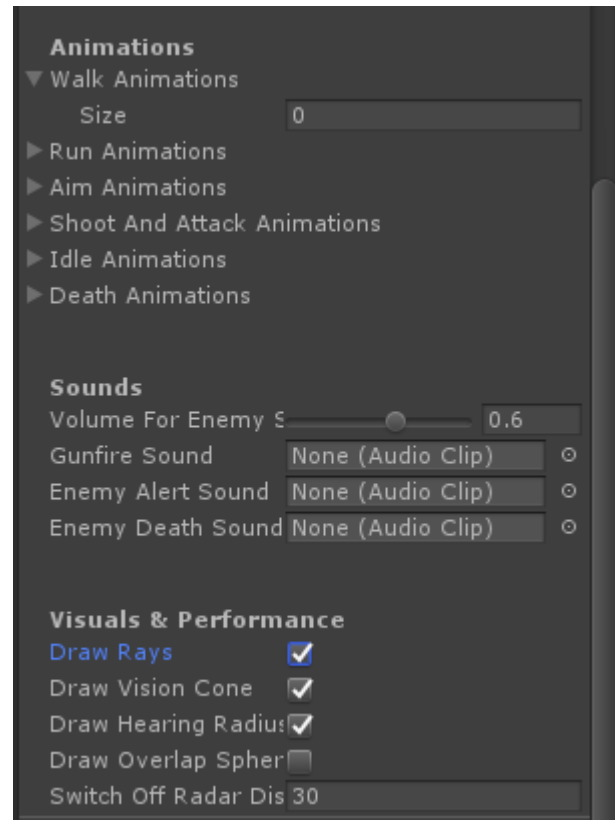
To manually add way points, please follow the steps below:

1. Select the AIEnemy shell (as created and defined above) to which you need to add patrol way points.
2. Right click on the shell and select 3D Object, then Sphere. This creates a unity primitive sphere at the center of the AI Shell transform (This new primitive shape defines your new waypoint, and you may freely change its position and scale to suit the needs of your game)
3. Next locate the script titled “WaypointIdentifier” (located in the Main AEA folder, under Scripts sub-folder) and add to your newly created waypoint. Adding this script is necessary if you want your waypoint to be recognized and registered by the AI system.
4. Repeat steps 1 – 3 above to add as many patrol way points as required.

NB: There is no need to disable mesh renderers or colliders for your manually created way points. The system automatically takes care of that.

Section C (Configuring AI)

Configuration of the AI parameters is done on the AIData script that is attached to the main AI shell as defined in the section above.



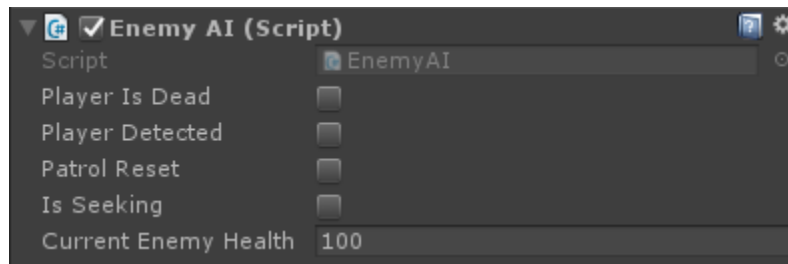
Below is a list of parameters contained in the script, along with descriptions of what they do.

PARAMETER	DESCRIPTION
Player	This is the Player or target gameObject that the enemy should find.
Walk Speed	Enemy speed while walking
Run Speed	Enemy speed while running
Max Patrol Wait Time	During patrol, maximum time to wait before moving to next patrol point
Max Attack Distance	Maximum distance to begin attack when player is detected
Min Attack Distance	Minimum distance to begin attack when player is detected
Enemy Behavior	Defines the enemy behavior. Patrolling way points or standing at a fixed position. Position and rotation will remain as initially set until an event occurs that requires the enemy to move or look
Random Patroller	If or not the patrolling enemy moves to way points at random
Sight Distance	Enemy can see as far as this distance
View Angle	Enemy can see as wide as this angle (in degrees)
Can Hear (Checkbox)	Enables/disables the enemy's hearing sensor
Hearing Radius	Enemy can hear as far as this distance in a circular fashion
Audibility Threshold	Enemy can hear target if it produces sound with volume equal to or greater than this value
Walk Animations	Provides slots to add walk animations for the AI entity
Run Animations	Provides slots to add run animations for the AI entity
Aim Animations	Provides slots to add aim animations for the AI entity
Shoot Animations	Provides slots to add shoot animations for the AI entity
Idle Animations	Provides slots to add idle animations for the AI entity
Death Animations	Provides slots to add death animations for the AI entity
AI Volume	Maximum volume of sounds that the AI agent produces

Gunfire Sound	The sound that the AI entity plays when shooting/Attacking
Enemy Alert Sound	The sound that the AI entity plays when it detects the player/target
Enemy Death Sound	The sound that the AI entity plays when its health reaches zero and the entity dies
Enemy Take Damage Value	Percentage of damage to take when enemy is attacked by player. Ignore if using another damage system
Player Take Damage Value	Percentage of damage to deduct from player when attacked by enemy (once for each attack strike). Ignore if using another damage system
Has Death Anim	If or not the enemy has a death animation
Death Spawn Items	Items to be spawned on enemy death.
Delay Before Destroy	Time taken before spawned items disappear
Draw Rays	If or not rays that the AI entity shoots should be shown in the editor (For debugging)
Draw Vision Cone	If or not the AI entity's cone of vision should be shown in the editor (For debugging)
Draw Hearing Radius	If or not the AI entity's circle of hearing should be shown in the editor (For debugging)
Draw Overlap Sphere	If or not the AI entity's overlap sphere should be shown in the editor (For debugging)
Switch Off Radar Distance	Turn off enemy radar when player is away by this value (save performance) if set value is less than both sight distance and hearing radius of this agent, the set value will be ignored since radar should not be off if player is within sight or hearing distance

NB: There are multiple slots for parameters such as animations. This is so that you can insert multiple clips and have the AI entity play one at random. Doing so ensures that the AI acts as random as possible every single time.

The EnemyAI Script



The “EnemyAI” script that is attached to the AI entity (the gameObject that is contained in the AI shell) is the main brain of the entire system. It contains public parameters for debugging during gameplay so that the system can be tested while in play mode. These parameters are listed below along with their descriptions

Parameter	Description
Player Is Dead	Simulates what happens if player/target is dead
Player Detected	Simulates what happens if player/target is detected by the AI entity
Patrol Reset	Restarts the AI Entity
Is Seeking	Simulates what happens if the AI entity is seeking the player/target. Usually when The AI entity hears the player/target
Current Enemy Health	The current health of the selected AI entity. This can be used in your health system to decrease enemy health, and eventually kill the enemy. Drag this to zero to simulate what happens when the AI entity dies.

The PlayerHealthManager Script

This script should be added to the player if this health system is to be used. The script contains a function called `killPlayer(){}.` This function gets called when the player health reaches zero and the player dies. It should be modified to suit your own game needs.

For questions and further inquiries, please contact us via email worqmedia@gmail.com