AI

cowsins

FPS ENGINE.

DOCUMENTATION

By Comrad Elmo

AI add-on for FPS Engine by Comrad Elmo.

Version 0.1.5

INDEX

Introduction

Introduction to the add-on:

This asset is an unofficial release developed by Comrad Elmo of an AI engine that's made for the FPS Engine by Cowsins. This documentation will provide you with the necessary information to effectively utilize this add-on.

License agreement:

Please note that this add-on is designed to work exclusively with the FPS Engine, a paid asset developed and owned by Cowsins. The FPS Engine is licensed as a paid asset and is protected by copyright laws, as well as Unity's Asset Store copyright terms and End User License Agreement (EULA). To utilize this add-on, it is mandatory to have a valid and licensed version of the FPS Engine.

Visit FPS Engine on the Unity Asset Store:

https://assetstore.unity.com/packages/templates/systems/fps-engine-218594

Please ensure that you have read and understood the applicable licensing and copyright terms before proceeding to use the FPS Engine and this add-on. Failure to comply with the terms outlined in the licenses and agreements may result in legal consequences and the revocation of your right to use the software.

Before Getting Started:

This asset was made using duplicates of some files made for the FPS Engine. Main assets (Player Controller etc) don't need to be modified for this asset to work. The Cowsins AI is made using a modified version of the Enemy.cs script (this will be changed as the asset updates) called EnemyAI.cs which is used within AI Health Manager.cs.

Note, this asset was made using Unity version 2021.3.13f1 and using the Universal Render Pipeline, the only custom textures used for this were the demo characters and the demo materials.

The actual Cowsins AI is viewable within the 'Demo' scene in Assets > Cowsins AI > Scenes > Demo. This is just a basic environment using the Shooter variant of the AI. Other prefabs are avaialable in Assets > Cowsins AI > Prefabs.

HOW TO INSTALL

In order for this asset to work, the 'Release' file must be dropped in the main Assets folder of your project. In order to stay updated for future updates, you need to ensure that the folder stays in that location.

After installation, ensure you have created an ‘Enemy’,‘Obstruction’ and ‘Player’ layer in order for the AI to function correctly. You will need to assign the ‘Player’ layer to the Player gameObject that has the key components in order for the AI to be able to see you. You may ALSO need to add the ‘Obstruction’ layer to the culling mask in the ‘Camera’ gameObject of the character controller in order to be able to see obstructions. You will also need to create a ‘BodyShot’ tag in order for colliders to register. On the prefabs, you may need to reassign the layers due to it missing what it requires.

CONTENT

In this section, we will explore the various variables used in the add-on and their respective functionalities.

The content section of this documentation is organized in a manner that mirrors the file structure of the FPS Engine add-on. This organization ensures easy navigation and a logical flow of information. Let's explore how the content is structured:

Assets/CowsinsAI/Scripts:

AIHealthManager.cs

CurrentState

WaitTime

UseWaypoints

Waypoints

WanderRadius

MinWanderDistance

MaxWanderDistance

SearchRadius

SearchAngle

Player

TargetMask

ObstructionMask

WaitTimeToSearch

Shooter

Projectile

FirePoint

ShooterAnimator

TimeBetweenAttacks

ShootDistance

InShootingDistance

Melee

MeleeDistance

MeleeAnimator

WaitBetweenAttack

CowsinsAI.cs

DamagePlayerMelee.cs

EnemyAI.cs

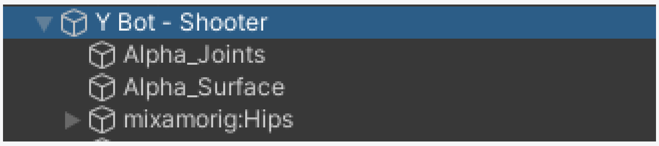
ProjectileDamage.cs

HOW TO CREATE AI

Old Setup

In order to create an AI, you need to think of several important things. If you are creating a AI with the ability to shoot, you need to create several animation states: 'Aiming Idle, Aimed Walking, Firing, Walk and Idle.' These animations are required in order for the script to work.

After that, you need to drag the model that you wish to use into the scene, unpack it and rename it to what you want. For the case of this documentation, I have named mine 'Y Bot - Shooter'. This is what I will be assigning all the scripts to.



After that, you need to assign the 'Cowsins AI, AI Health Manager and Nav Mesh Agent' to the GameObject.

You also need to bake a NavMesh (Video by Code Monkey) in order for the AI to be able to walk around on the ground.

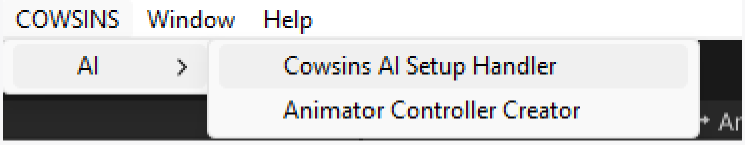
AI Health Manager

Within AI Health Manager, you need to set the maximum health along with maximum shield (if you wish to use it). You also need to assign sliders for both of these. In order to obtain these, you can either make your own or get them from either the ‘Target’ prefab or one of the AI prefabs available. Make sure that they are assigned under the parent GameObject. After that, assign the variables to the correct location.

You will also need to assign a ‘Damage Pop Up’ otherwise it will throw an error. You can get one by searching ‘DMGPopUp’ in the Project window. The color variables will also need to be assigned. The best way of doing this is by using the ‘Eyedrop’ button on the color and clicking the color in the scene view. After that, you are able to continue to ‘Cowsins AI’.

Simple Setup

Since Cowsins AI update 1.1.5, there is an easier way to create AI within Cowsins AI. In order to do so, proceed to the following:



Within here, you can allocate:

- The name of the AI
- AI GameObject
- Tag of the AI
- Layer of the AI
- Whether it will use Ragdoll
- AI Type (Shooter, NPC)
- Enemy Type (Shooter, Melee)
- Shooter Type (Hitscan, Projectile)
- Bullet Trail GameObject
- Hit Mask
- Movement Type (Waypoint, Random, Idle)

When complete, the AI will automatically be configured to your specification. After that, you can fine tune all settings to what you want which includes the NavMeshAgent settings and Enemy AI Health settings.

Cowsins AI

‘Variables’ Tab

Within ‘Cowsins AI’ a lot of variables need to be set in order for the AI to function correctly. All required variables are located in the ‘Variables’ tab. First, ‘Wait Time’; this variable defines how long the AI should wait until going to the next waypoint or when it should move again.

Next, the ‘Use Ragdoll’ variable is available. I will not be going into detail on how to make one of those. Many great resources are available online where they go into detail on how to make one. As of 1.1.0, ragdolls only fall on the floor and do not apply damage depending on where they’re shot/where an explosion occurred and the force amount.

The ‘Dumb AI’ variable defines whether the AI has the ability to move or stand stationary and NOT attack the player. When this variable is disabled, a lot of variables are available to set.

With the ‘Dumb AI’ variable being disabled, ‘Use Waypoints, Wander Radius, Minimum Wander Distance and Maximum Wander Distance’ need to be assigned.

‘Use Waypoints’ defines whether the AI shall use waypoints in order to idle. If ENABLED, a list is available where you are able to assign empty GameObjects as positions. The AI will start at ‘Element 0’ and continue until the end of the list. Once the end is met, it will repeat the path over and over again. With the variable being DISABLED, it will randomly wander around the NavMesh. You can change the other variables mentioned above to what you wish.

Next, ‘Player Searching Variables’ are available. Here, you assign ‘Search Radius, Increase Sight on Attack, Search Angle, Target Mask, Obstruction Mask and Wait Time to Search.

‘Search Radius’ defines the radius in which the FOV can see you. This works alongside ‘Search Angle’ which defines the actual FOV (Field of Vision) where the AI can see you in. The AI will only react to the ‘Target Mask’ (which is Player) in which the rest of its functions can work.

The ‘Obstruction Mask’ is the layer in which the FOV **CANNOT** see the ‘Target Mask’ through. This is useful for things like walls and objects around the map.

The ‘Wait Time to Search’ variable is what defines how long the AI will spend trying to find the player after losing sight of them.

‘Combat’ Tab

Within this tab, you will initially find two bools; Shooter and Melee. Only **ONE** of these bools may be active at a time to avoid errors during the game. In this case, I will be using the ‘Shooter’ mode. The Melee and Shooter modes have similar variables to assign.

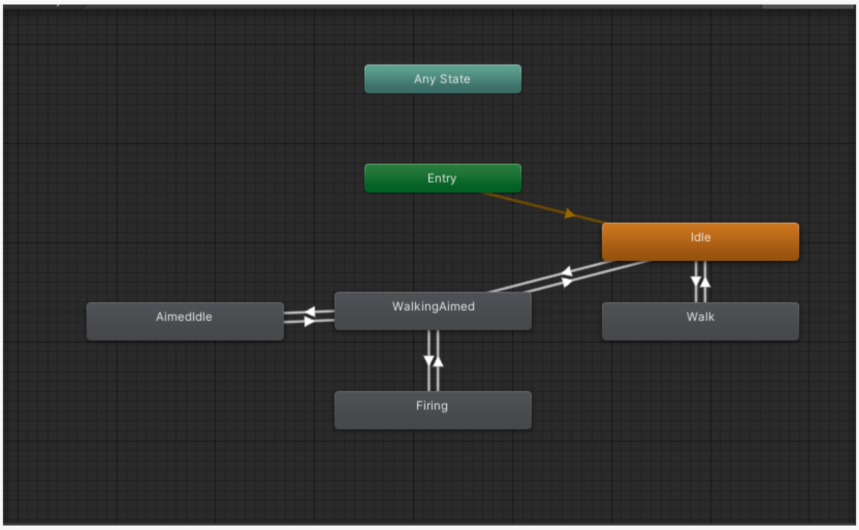
The first variable is ‘Type’. This defines how the shooting will work. There’s ‘Projectile and Hitscan’. Using ‘Projectile’, you need to assign a GameObject which has the ‘Projectile Damage’ script attached along with a Rigidbody and Collider. The ‘Hitscan’ mode requires several things assigned. You first need to have a ‘Bullet Trail’. One for testing purposes can be found in: ‘Assets > Cowsins AI > Prefabs > BulletTrail.’

You also need to assign a spread amount. This amount defines how much the bullets, well, spread. Making this 0 makes it so it is in a straight line. You also need to assign a ‘Hit Mask’ which is also the ‘Player’ layer.

After that, the ‘control’ variables need to be set. That includes a ‘Firepoint, Shooter Animator, Shoot Distance and Time Between Shots’. The ‘Firepoint’ needs to be assigned where the gun shoots from. (In order for the gun to work, ensure that you attach the weapon GameObject to the ‘hand’ bone in the character so it follows).

Old Animator Creation

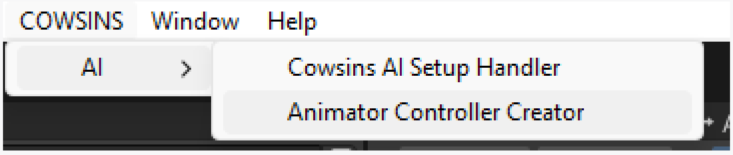
Next, the shooter animator needs to be assigned. This is where you create the animator controller that is used. A template controller is available in: ‘Assets > Cowsins AI > Animations > Core > ShooterTemplate.’ Duplicate this and rename it to what you want. Go into it and assign the animations necessary.

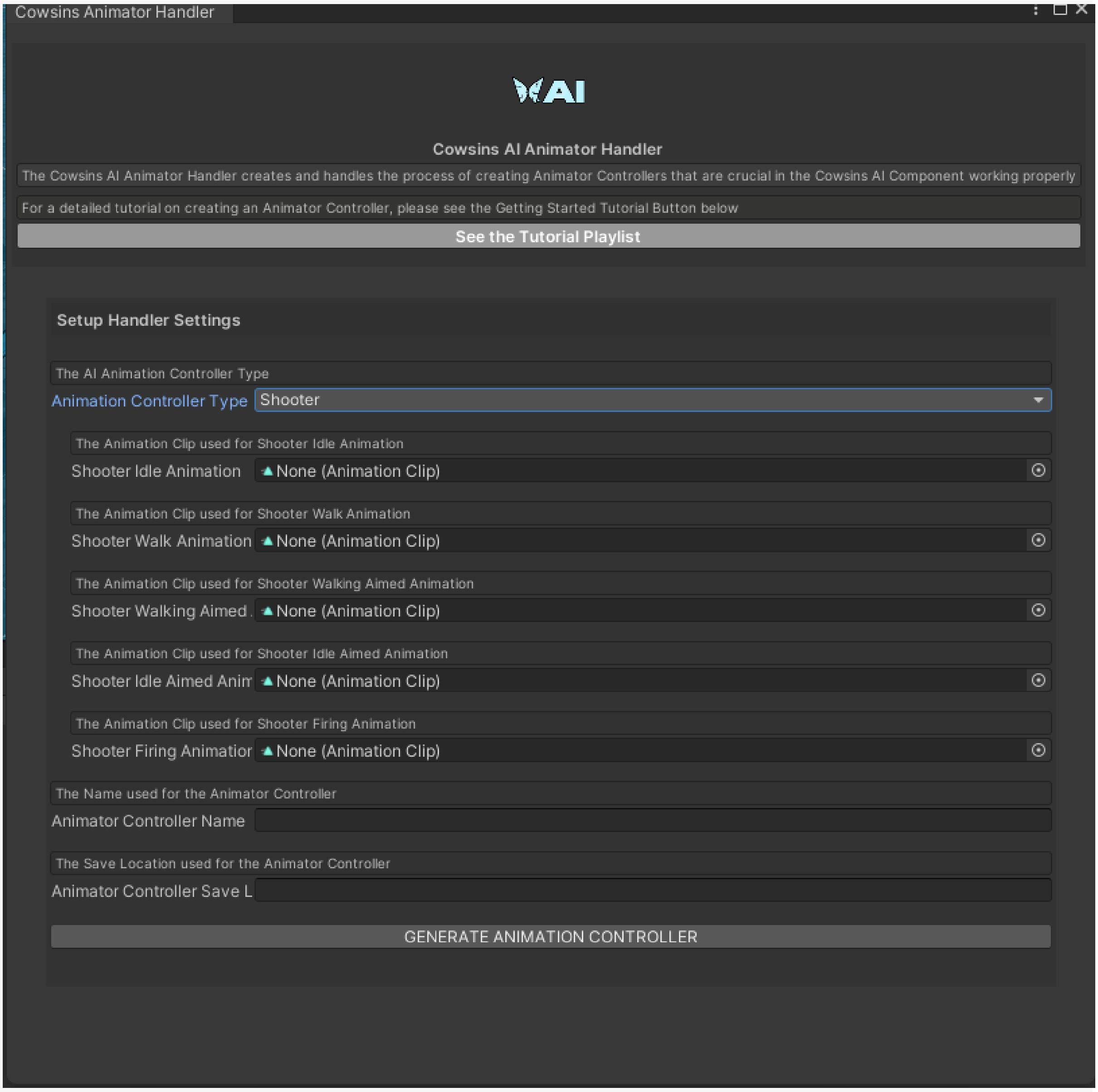


After that, drag the animator controller as a component of the AI and then assign it to the 'Shooter Animator' variable.

New Animator Creator

As of version 1.1.5, you are able to make animators easily using the same method you create AI. You can create them in the following location:





Cowsins Animator Handler

Cowsins AI Animator Handler

The Cowsins AI Animator Handler creates and handles the process of creating Animator Controllers that are crucial in the Cowsins AI Component working properly

For a detailed tutorial on creating an Animator Controller, please see the Getting Started Tutorial Button below

Setup Handler Settings

The AI Animation Controller Type
Animation Controller Type: Shooter

The Animation Clip used for Shooter Idle Animation
Shooter Idle Animation: None (Animation Clip)

The Animation Clip used for Shooter Walk Animation
Shooter Walk Animation: None (Animation Clip)

The Animation Clip used for Shooter Walking Aimed Animation
Shooter Walking Aimed .: None (Animation Clip)

The Animation Clip used for Shooter Idle Aimed Animation
Shooter Idle Aimed Anir: None (Animation Clip)

The Animation Clip used for Shooter Firing Animation
Shooter Firing Animatior: None (Animation Clip)

The Name used for the Animator Controller
Animator Controller Name:

The Save Location used for the Animator Controller
Animator Controller Save L:

Within this, you're able to allocate animations for all Shooter, Melee and NPC types that will automatically be created for you (no more duplicating animators).

Cowsins AI Continued

The 'Shoot Distance' is how far the AI will shoot you from. A debug view of this can be found in the 'Debug' tab.

Lastly, you have the 'Time Between Shots' which defines how long it will wait between shooting the player.

That is all that needs to be done to create an AI. Enjoy and remember to donate to my Gumroad in order to help me continue development of this asset.

FAQ

The AI Can See the Player Through Walls
Ensure that the obstacles that you wish to have as walls have the ‘Obstruction’ layer on it.

The AI is not moving
Ensure that the surface the AI is placed on has a Navmesh baked onto it in order for this asset to work correctly.

The AI is freaking out
If the AI’s body is ‘freaking out’ and you’re **not** using ragdolls, ensure that the ‘Use Ragdolls’ bool in the ‘Cowsins AI’ is **disabled** or this issue will continue.

Custom animations for the AI are not working correctly
In order for the AI’s animations to function correctly, refer to a demo prefab (Assets > Cowsins AI > Animations > Core > Shooter) and look at how the animation tree is layed out. This layout is required in order for the script to register movements correctly.

Any issues
Please do not refrain from asking for support about the asset. Remember, Cowsins are **not** the creators of this asset so please do not ping any other staff. If there are any bugs that you encounter, run ‘/reportbug’ in the ‘bot-commands’ channel where Comrad Elmo will look further into this issue.