



Hello!

First off, thank you very much for buying Shooter AI and supporting us in continuing to make SAI and more products.

This is the manual for Shooter AI 2.x cycle and will guide you from the basic setup, right up to fine tuning the details to make your shooter game have awesome AI.

Be sure to watch all of our video tutorials on YouTube:

<https://www.youtube.com/playlist?list=PLrzO1o8MansJl9Qqw-dvh6uCtqB7rCaVd>

Also, you can get quick support by going on to our website:

<http://nikita-makarov.wix.com/gateway-games#!support/c21nl>

Thank you very much,  
Gateway Games

Quick Setup .....	3
Detailed Component Info .....	9
Brain .....	9
AI Stats .....	9
AI Speeds .....	10
Enemy Data .....	10
Emotion Data.....	11
Engagement Scripts.....	11
Weapon Manager .....	12
Multiple Weapons .....	12
Secondary Weapon Data .....	12
Health Manager .....	13
Health Regions .....	13
Weapon.....	14
IK Setup.....	16
General Setup .....	16
Hand Selection .....	16
Hand Position .....	16
Weapon Position .....	17

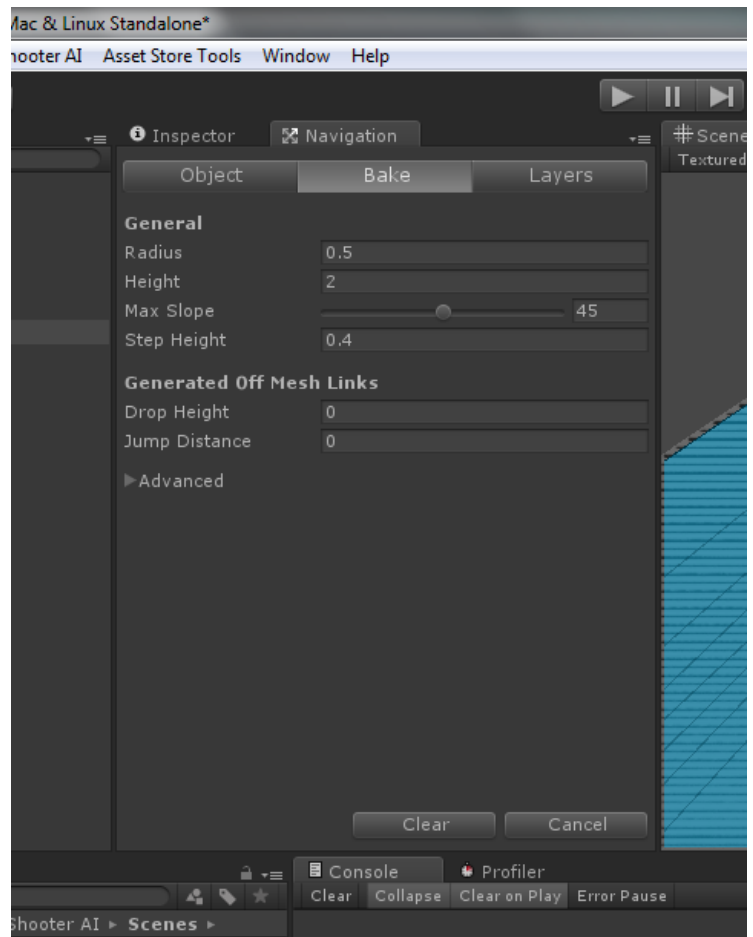
# Quick Setup

This is the first part of the manual for Shooter AI and will cover the basics on how to setup an AI Character. Bold sentences are instructions, and everything not in brackets are buttons.

## 1. Scene Setup

First we shall setup the scene so that it's ready for the AI. By this step you should have already finished your environment.

**Window -> Navigation -> (Configure settings as you want, default is usually good)**  
**-> Bake**



## 2. Weapon Setup

### Preparations:

#### I) Empty magazine setup (Optional)

Now we setup the empty magazine object. This is a model of the magazine with a rigidbody.

As with the bullet, we provide a prefab under

“**ShooterAI/Prefabs/Magazines/Magazine1.prefab**”. We recommend you modify it for your own game.

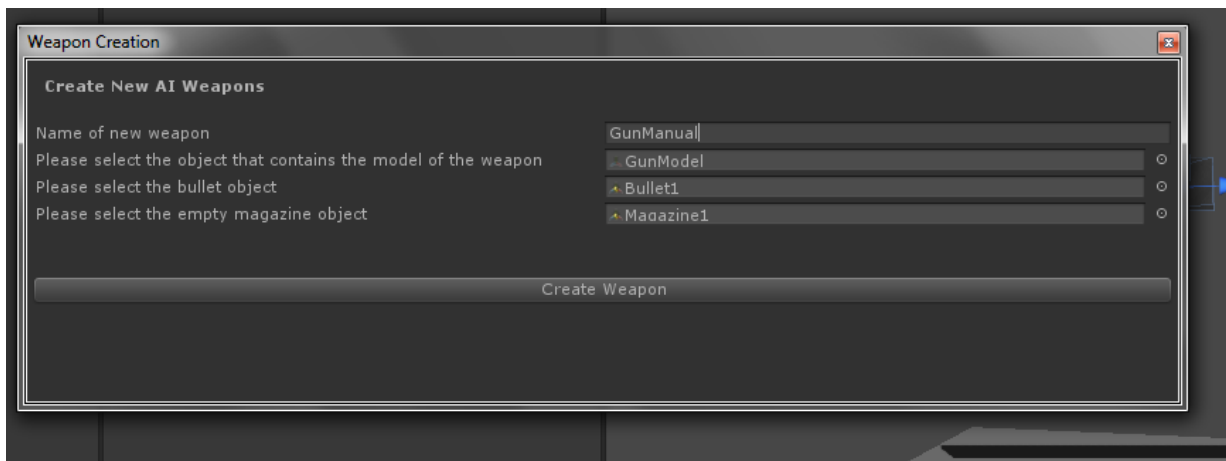
#### III) Weapon Model

For the weapon model you can use any Gameobject.

### Main Setup:

The setup is extremely simple, and uses the objects that we prepared in the last few steps.

**Shooter AI -> Create New Weapon -> (Type in name for your weapon) -> (Drag and drop in the weapon model) -> Create Weapon**

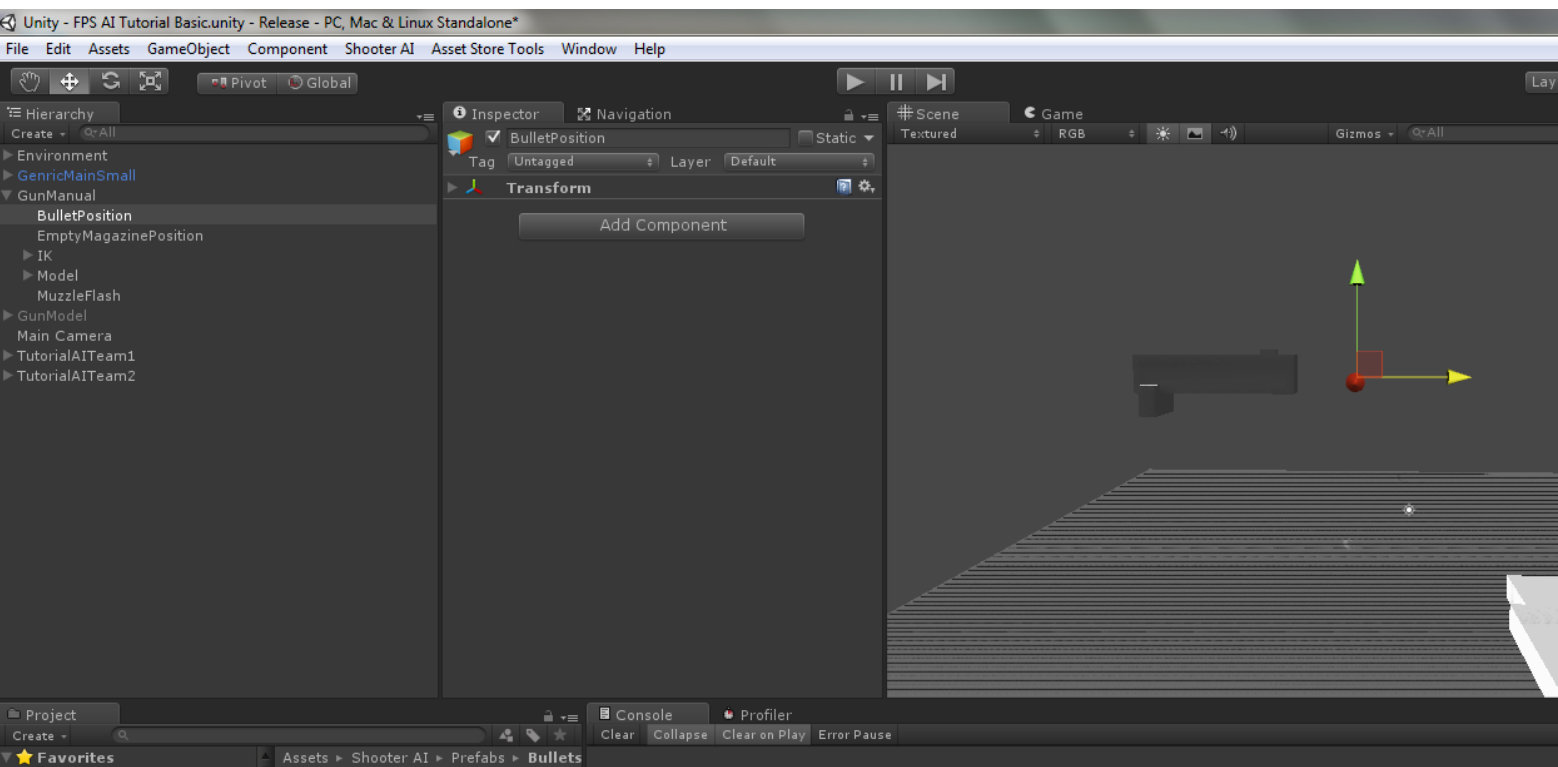


Now your weapon is almost fully setup and we just need to adjust some values.

First we shall setup some values. **Once you created the weapon, it is ready to be used, though without adjustment it might produce strange effects, such as the muzzle flash is off to the side.**

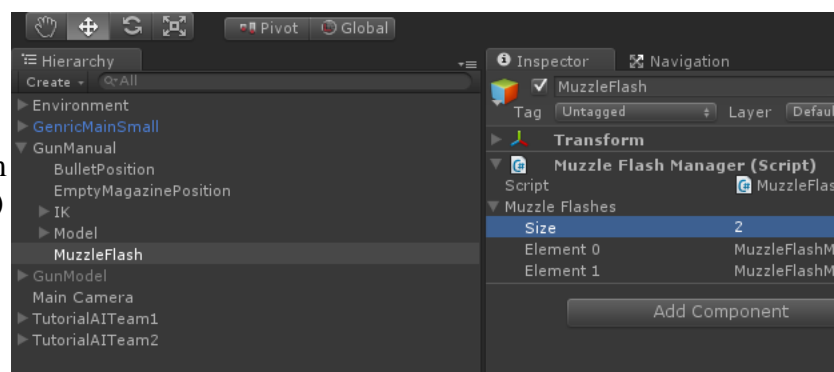
You can overview all parameters by **(Click on the gun object) -> (Open script “Gateway Games Weapon”)**. The parameters are grouped logically, and should be self-explanatory. If you need more details on a single parameter, open the script and find the comments for the variable.

Firstly we will configure the bullet creation position. To do this **(Click on the gun object) -> (Open the heirarchy) -> (Click on “Bullet+Muzzleflash” object) -> (Position the object to where you want the bullets and muzzleflash to come out)**



Now we configure the position of where the empty magazine flies out, the position of IK (where the left and right hand grab the gun) and where the muzzle flash goes, using the same principle but with their respective Gameobject.

If you want to use custom flashes and/or choose random flashes, **(Click on the gun object) -> (Open Heirachy) -> (Click on “MuzzleFlash” object) -> (Open “Gateway Games Muzzle Flash Manager” script) -> (Open array called “Muzzle Flashes”) -> (State the amount of flashes) -> (Drag and drop the flash prefabs into the slots)**



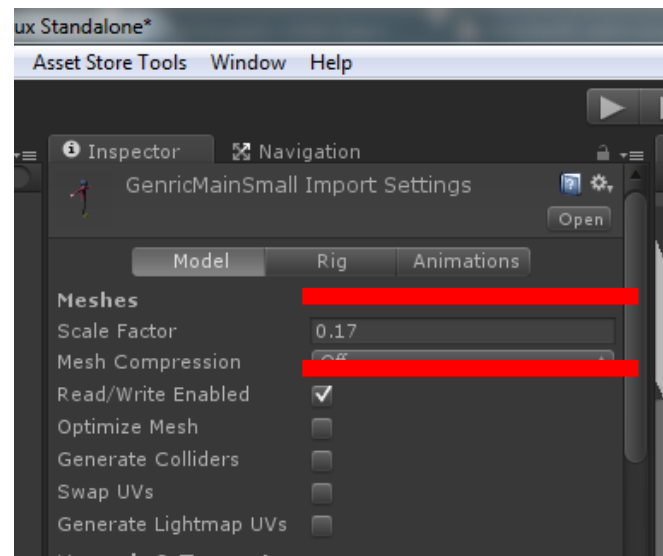
### 3. AI Character Setup

Now that we have finished setting up the gun, we can make the character itself.

#### Preparations:

We first have to setup the model with the ragdoll.

First you should download and import a character model into Unity. Mixamo has tons of ready made character models that are already Mecanim rigged and fully ready. **Because of a bug in Mecanim, all scaling has to be applied directly to the model!!!!**

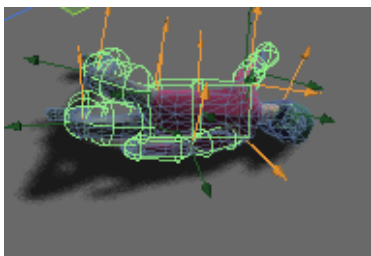


Once we're happy with the model and its properties, **drag and drop the model into the scene.** After that create a ragdoll object out of it. Here are some good tutorials, or look into Unity's documentation on it:

[https://www.youtube.com/watch?v=dCwNaE\\_eVsM](https://www.youtube.com/watch?v=dCwNaE_eVsM)

<https://www.youtube.com/watch?v=9Ehf6UtQp2I>

Once you've setup the ragdoll, test it in play mode and see if it tumbles as wanted.

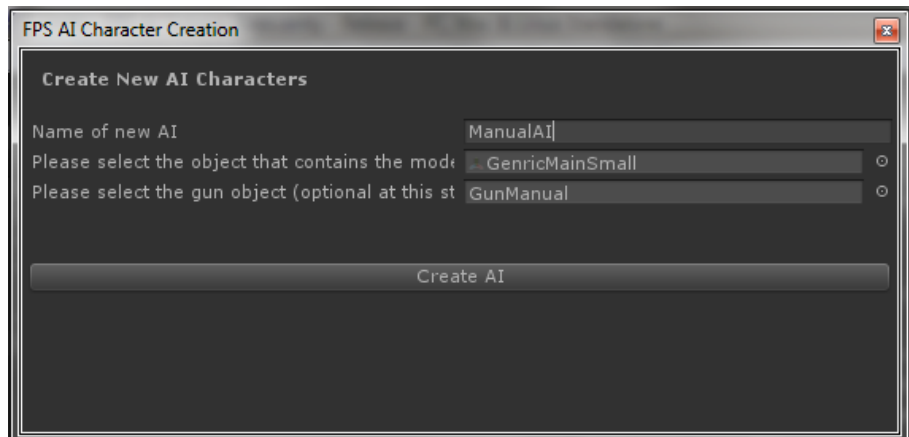


For the IK to work, you need to rotate each hand joint part slightly in the direction that it will be able to rotate when holding its weapon. We have videos that demonstrate how to do this exactly.

Now that the model is ready we can proceed to create the character itself.

#### Character Setup

**Shooter AI -> Create New Character -> (Type in name for new character) -> (Drag and drop in the object with the model+ragdoll) -> (Drag and drop in the gun object we made earlier) -> Create AI**



We're almost finished.

As with the gun, **the character's default parameters are ready-to-go**. If you want to view them, **(Click on the AI Character object) -> (Open script "Gateway Games Brain")**, and then you can view any property that influences the AI.

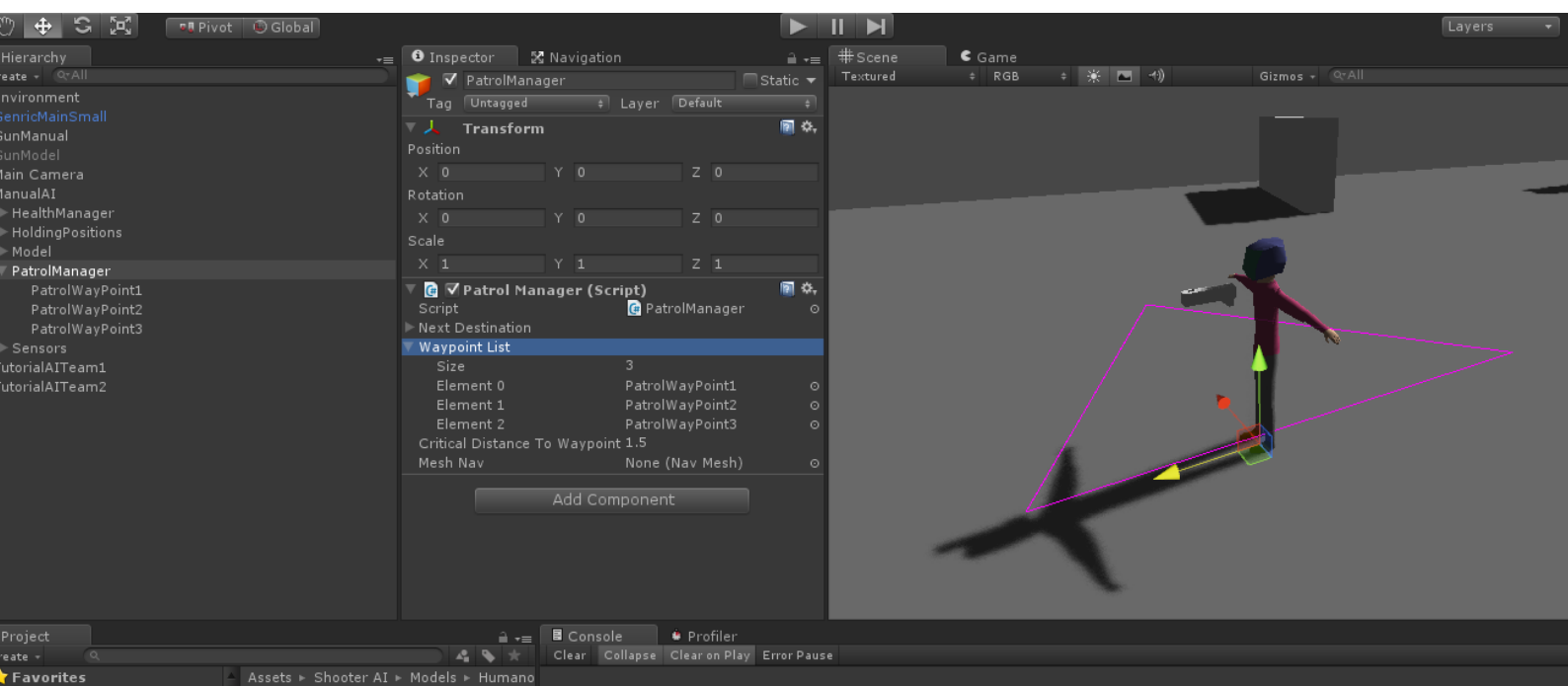
You may want to do some modifications to the gun e.g. Scaling. For that **(Click on main character) -> (Open its hierarchy) -> (Open "DefaultHoldingPostion" hierarchy) -> (Click on gun object)**

Finally we'll setup our waypoints and then we will be finished!

**(Open the main character and its hierarchy) -> (Open "Patrol Manager" hierarchy) -> (Move each waypoint to the AI should go) -> (Click on "Patrol Manager" script inside "PatrolManager" object to see the patrol path)**

If you want to use something other than the standard 3 waypoints, just **(Open "Patrol Manager" script) -> (Open "Waypoint List" array) -> (Resize array and change out objects)**

NOTE: if your waypoint object is large, you should increase the variable "Critical Distance To Waypoint", or else the AI might never get to the waypoint.



You have finished! After a practising it a couple of times, it is possible to make full AI in under 5 minutes! Have fun!

Also watch our YouTube tutorials on more visual documentation.

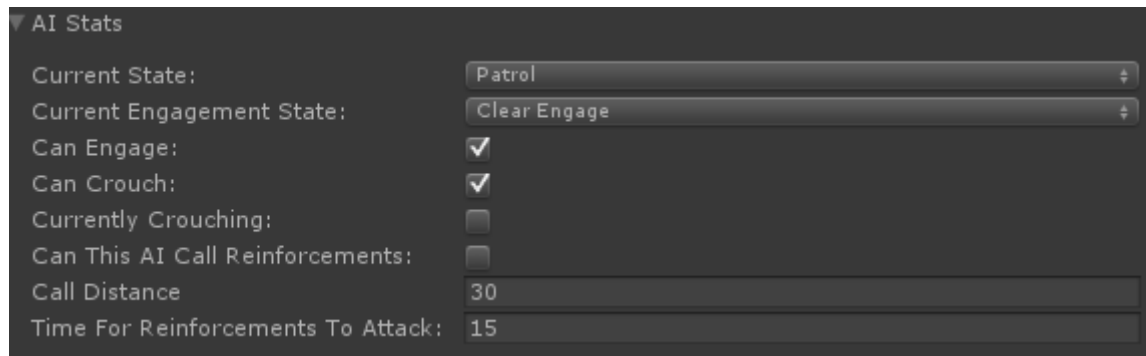


## Detailed Component Info

### ***Brain***

The brain serves as the nerve centre of the AI and all of its decisions are made here.

### **AI Stats**



A screenshot of a software interface titled "AI Stats". It contains several configuration options for an AI component. The options are listed on the left, and their current values or states are shown on the right. The interface has a dark, semi-transparent background.

AI Stats	Value/State
Current State:	Patrol
Current Engagement State:	Clear Engage
Can Engage:	<input checked="" type="checkbox"/>
Can Crouch:	<input checked="" type="checkbox"/>
Currently Crouching:	<input type="checkbox"/>
Can This AI Call Reinforcements:	<input type="checkbox"/>
Call Distance	30
Time For Reinforcements To Attack:	15

Current State - The current state of the AI. This is automatically determined in runtime.

Current Engagement State - The tactic the AI is currently trying to use. Calculated in runtime.

Can Engage - Whether the AI can engage or not.

Can Crouch - Whether the AI is allowed to crouch or not.

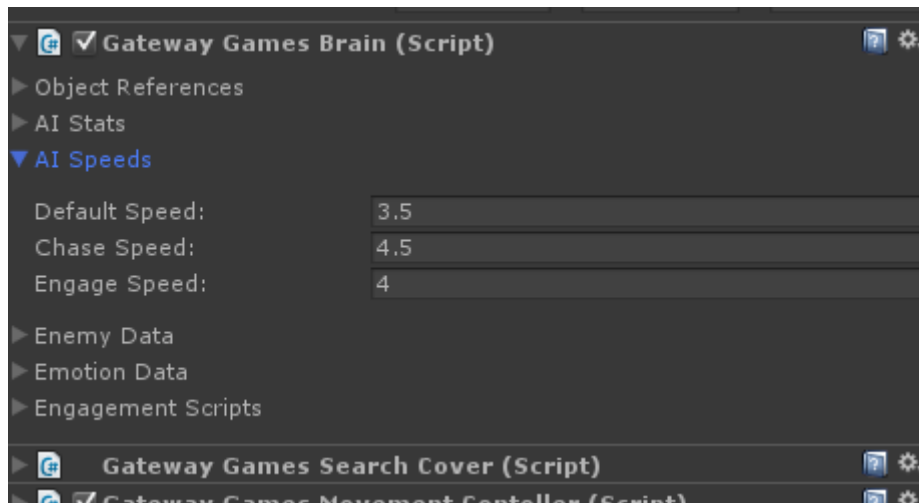
Currently Crouching - Whether the AI is currently crouching. Determined in runtime.

Can This AI Call Reinforcements - Whether this AI can call reinforcements

Call Distance - How far away can this AI call for reinforcements

Time For Reinforcements To Attack - Once called for reinforcement, how long should the AI attack the given enemy before going back to normal mode.

## AI Speeds

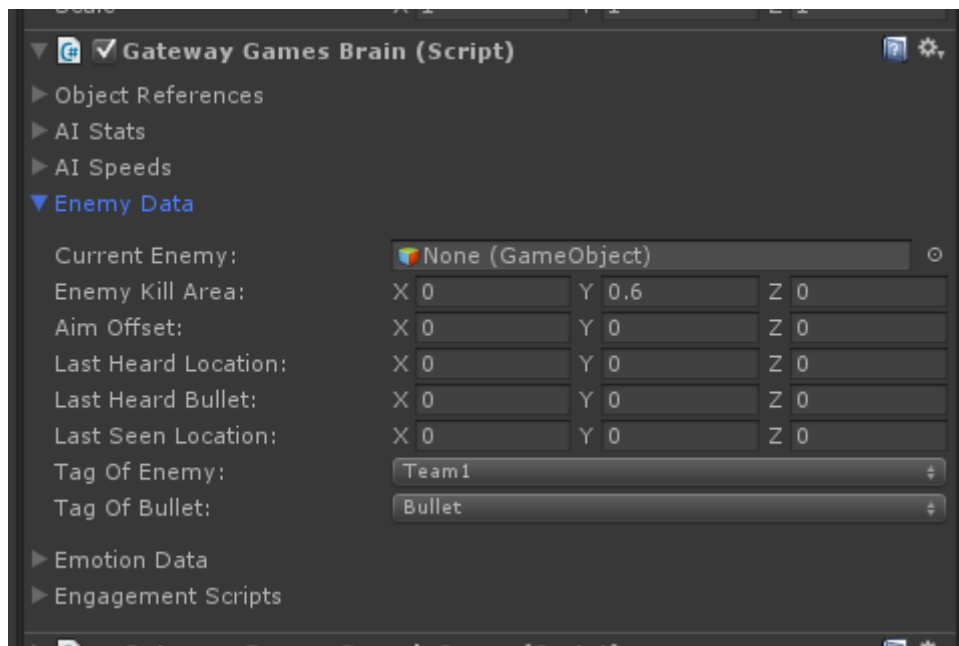


Default Speed - The default speed of the AI.

Chase Speed - The speed of the AI when chasing.

Engage Speed - The speed of the AI when engaging.

## Enemy Data



Current Enemy - The current enemy object. Determined in runtime.

Enemy Kill Area - The area where the AI should aim in the enemy.

Aim Offset - Offset applied to aiming. Determined in runtime.

Last Heard Location - The location where the AI last heard the enemy. Determined in runtime.

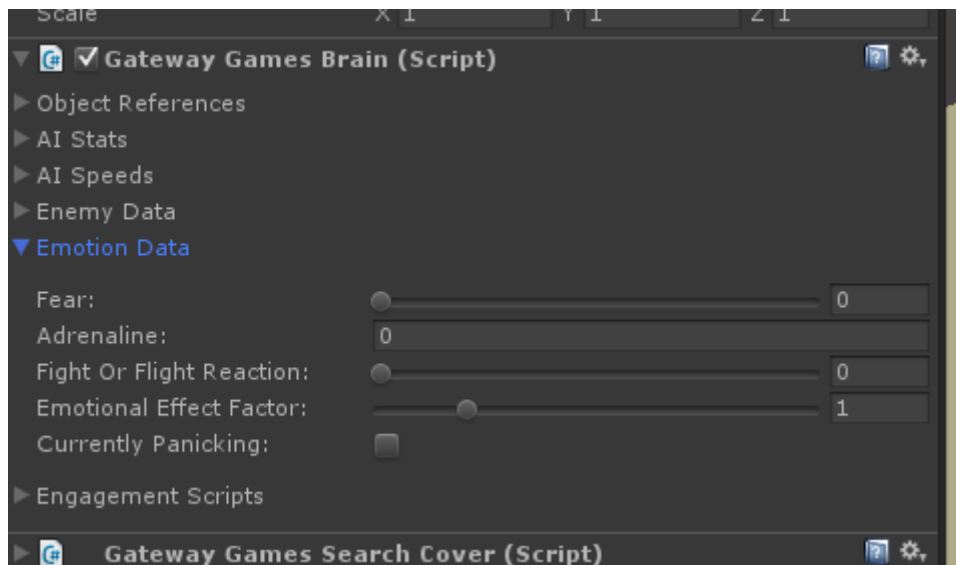
Last Heard Bullet - The location where the AI last heard a bullet. Determined in runtime.

Last Seen Location - The location where the AI last saw the enemy. Determined in runtime.

Tag of Enemy - The tag of the enemy.

Tag of Bullet - The tag of bullet objects.

## Emotion Data



Fear - The amount in percent of fear. After 80%, the AI panics.

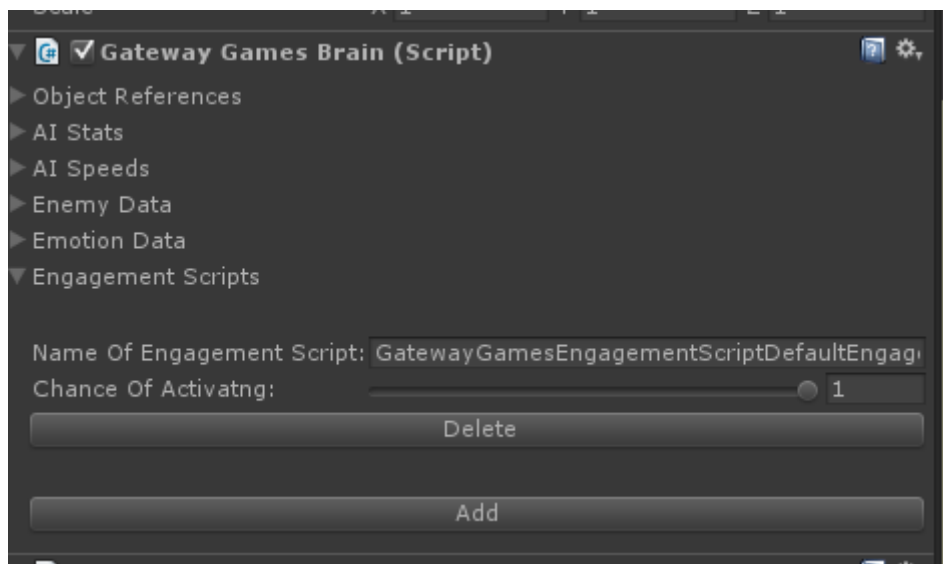
Adrenaline - The amount of adrenaline in the AI.

Fight or Flight Reaction - In decision making, what are the chances the AI will fight.

Emotional Effect Factor - How much do emotions effect the AI.

Currently Panicking - Whether the AI is currently panicking. Determined in runtime.

## Engagement Scripts

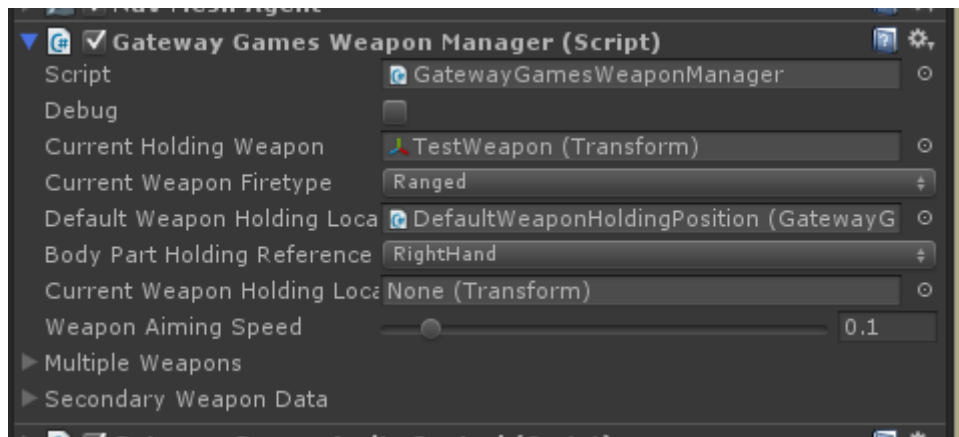


Each engagement script that you show here has a chance of being executed, and thus determines whether the AI will fight or stand down.

Engagement scripts must send a message to the brain either "EngageEnemy" or "StandDown".

You can add/delete engagement scripts with the "Add" and "Delete" keys. The name of the engagement script is the script name. The "Chance Of Activating" determines the percentage chance of the AI using this script upon seeing the enemy.

## Weapon Manager



Debug - Whether to show debug data.

Current Holding Weapon - The current weapon that the AI is holding.

Current Weapon Firetype - What type of weapon the AI is holding. Determined in runtime.

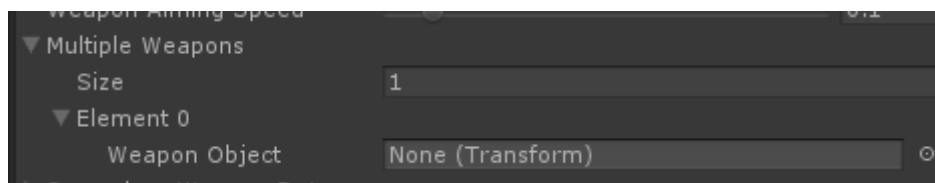
Default Weapon Holding Location - Where (in the hierarchy) the AI is to hold the weapon.

Body Part Holding Reference - What to use as reference of weapon position.

Current Weapon Holding Location - Where the AI is currently holding the weapon. Runtime.

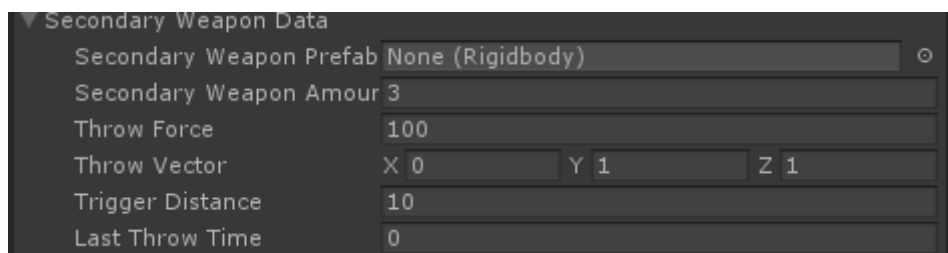
Weapon Aiming Speed - How quickly the AI yields the weapon.

## Multiple Weapons



The multiple weapons array determines all weapons the AI can use. The currently holding weapon has to be determined through a custom script. The API can be directly accessed in the GatewayGamesWeaponManager script.

## Secondary Weapon Data



Secondary Weapon Prefab - Grenade object.

Secondary Weapon Amount - The amount of grenades.

Throw Force - Throw force to apply to throw the grenade.

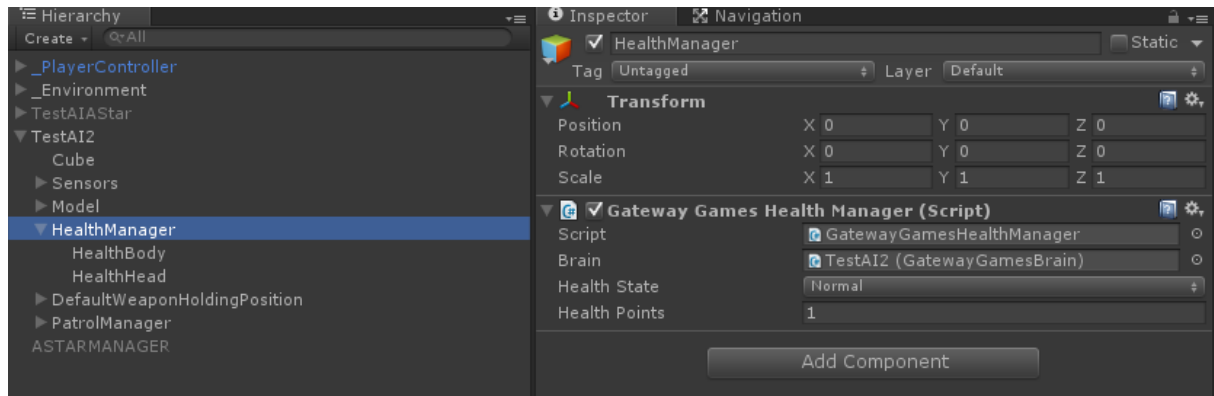
Throw Vector - Where the relatively throw the grenade.

Trigger Distance - Within what radius can grenades be used.

Last Throw Time - Time the last grenade was thrown. Determined in runtime.

## Health Manager

The HealthManager is a child of the main AI and holds all health related data.



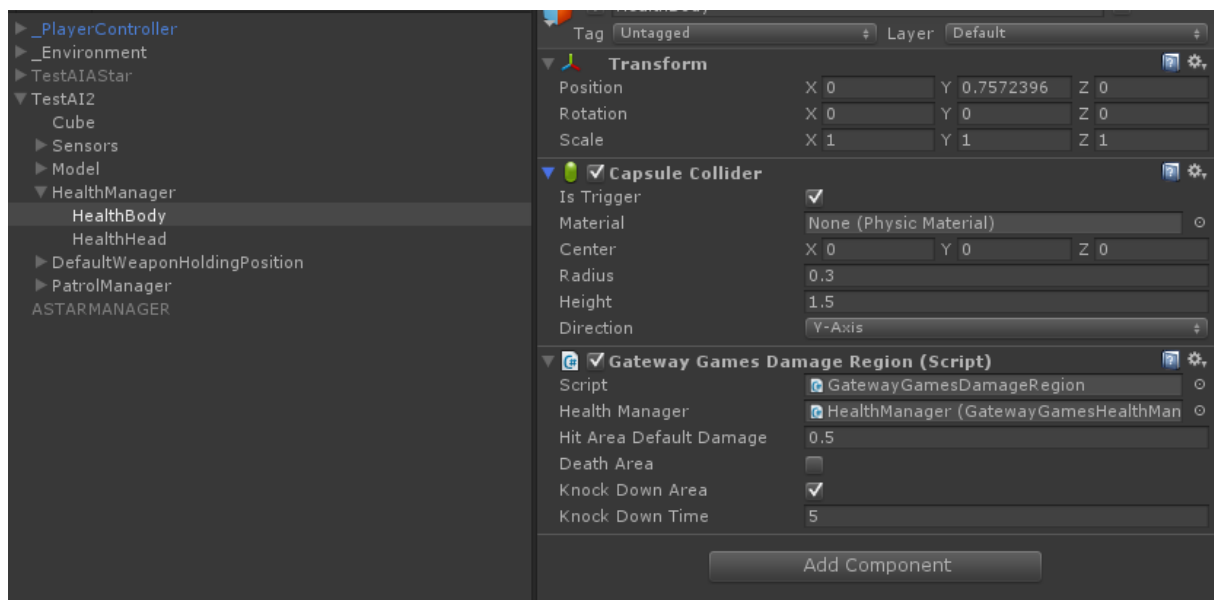
Brain - Reference to brain script.

Health State - The health state of the AI. Determined in runtime.

Health Points - Amount of health points the AI has.

## Health Regions

Health regions are children of the HealthManager and hold data of where the AI can get hit and how it affects it.



Collider - Has to be set to a trigger. Can be any type.

Health Manager - Reference to the health manager.

Hit Area Default Damage - How much damage is applied if this area is hit.

Death Area - Whether the AI dies instantly if this area is hit.

Knock Down Area - Whether the AI gets knocked down if a bullet hits this region.

Knock Down Time - If the bool above is true, then how long should the AI be knocked down.

## Weapon



Holding AI - The AI currently holding this weapon. Determined in runtime.

Bullet Creation Position - Where to create new bullets.

IK Left Hand - Reference to where the left hand should go.

IK Right Hand - Reference to where the right hand should go.

Muzzle Flash Manager - Reference to where muzzle flashes should be created and managed.

Debug - Whether to provide debug info.

Being Held - Whether this weapon is currently being held. Determined in runtime.

Weapon Type - What type of weapon this is.

IK Type - How should this weapon be held.

Weapon Shoot Method - What type of bullet modelling should be used.

Projectile Prefab - (If Projectile) Prefab of bullet.

Projectile Force - (If Projectile) force with which to shoot it.

Ammo In Magazine - How much ammo is in the current magazine.

Ammo Per Magazine - How much ammo is in each magazine.

Amount of Magazines - How many magazines for this weapon does the AI have.

Allowed To Shoot - Whether this weapon is allowed to shoot.

Time Between Shots - How much time elapses between each shot.

Recoil Vector - How the recoil will go.

Recoil Amount - How much recoil.

Damage Model - Whether to apply damage based on where the bullet hit, or uniformly.

Damage - (If Global damage model) Amount of damage to apply to the enemy.

Reloading - Whether the weapon is currently reloading. Determined in runtime.

Reload Time - How long the weapon reloads.

Empty Magazine - Object to create to represent an empty magazine after reloading.

Empty Magazine Throw Vector - How to throw the empty magazine.

Empty Magazine Throw Force - How hard to throw the empty magazine.

Impact Effect - What object to create on impact.

Bullet Shot Sounds - Possible bullet shot audio sounds.

Melee Default Length - Default length of melee animation in seconds.

Melee Attack Distance - From which distance to attack with melee.

# IK Setup

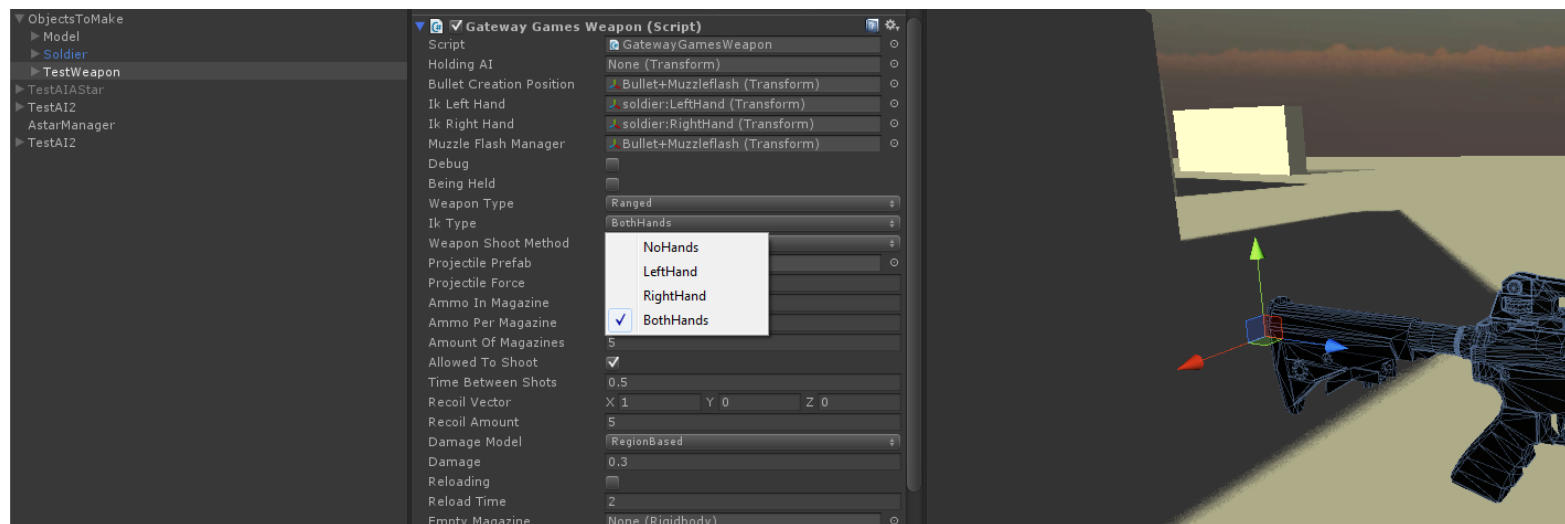
Inverse Kinematics makes the AI hold their weapons. In most AI solutions, this is implemented through animations, but we decided that it would take too much time for each client to make custom weapon animations, so we went procedural. [The IK system in SAI 2.x is designed by Pärtel Lang, who is a guru in procedural animations, and has AAA quality products available on the Asset Store.](#)

## General Setup

The main parameters for IK can be adjusted on the weapon itself.

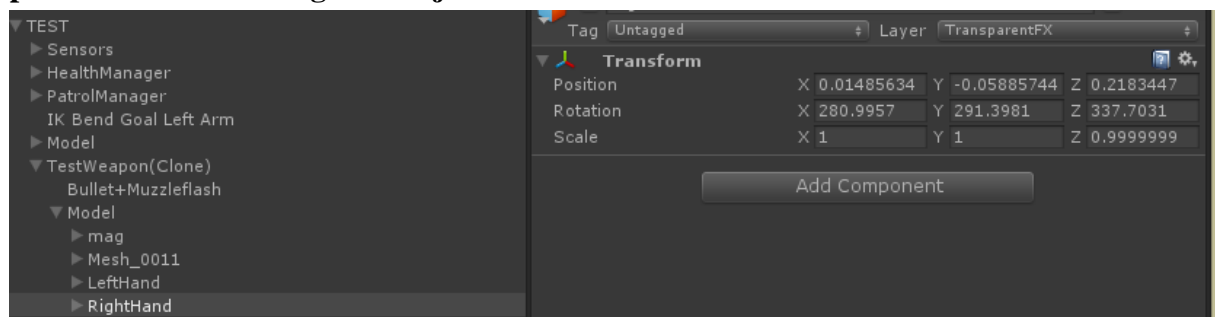
## Hand Selection

To select which hand the AI should hold this weapon, **Select your weapon -> Open the script "Gateway Games Weapon" -> In "Ik Type", select the hand that should hold it**



## Hand Position

Usually, the default hand placement set on the weapon is off for your custom weapon. To adjust where and how the hands hold the weapon, **Select your weapon -> Open the hierarchy -> Open the children of the "Model" object -> Select, move and rotate the "LeftHand" and "RightHand" objects -> (Optional) Open their respective children and position individual fingers and joints.**





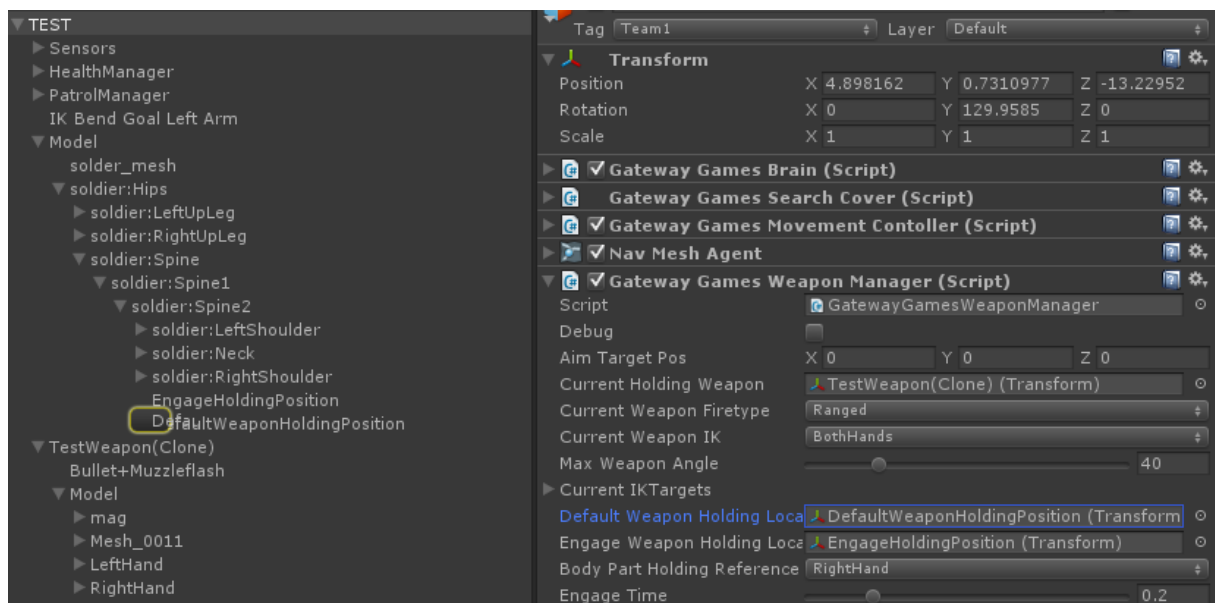
By positioning the objects the way you want, Shooter AI recognizes and calculates the best way to hold the weapon the way you want.

Sometimes it takes quite a bit of practise to get the right way, and we recommend positioning the IK objects in runtime, noting their positions/rotations, then stop and apply the changes and finally save.

## Weapon Position

The weapon position is totally determined by reference objects depending on the current state of the AI. There are 2 objects: Normal and Engage. The Engage reference object is used when the AI is attacking, chasing or investigating. Normal is used in all other situations, such as patrolling.

To change their positions/rotations, **Select your AI -> Open the script "Gateway Games Weapon Manager" -> Click on the "Default Weapon Holding Location" or "Engage Weapon Holding Location" to find the objects in hierarchy**



**Then by assigning the position/rotation that you want, the weapon will be held there during the respective state.**