



Hello there!

This is the manual for the Shooter AI developed by Gateway Games. Shooter AI is a dynamic, emotion based artificial intelligence created for quick and easy deployment.

We have a number of video tutorials available on YouTube:

<http://nikita-makarov.wix.com/gateway-games#!help/cj2w>

And you can get very quick support through our support page:

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

The first part of this documentation is a quick setup, followed by a detailed view of all properties and finally covering the basics of writing engagement scripts.

Thank you very much,
Gateway Games

Quick Setup.....	3
1. Scene Setup	3
2. Weapon Setup	4
3. AI Character Setup	6
Preparations.....	6
Character Setup	7
Detailed Property Overview	10
Player properties.....	10
Main Properties	10
Health Engine	14
Gun Properties.....	15
Melee setup.....	16
AI Character Setup	16
Melee Animations	16
How To Write Own Engagement Script	18
Ultimate FPS with Shooter AI	20
Team Control	21
Team Creation and Management	21
Team Creation.....	21
Team management.....	22
Behaviour and Event Management	22
Team Strategy	23
Multiple Teams	24
Logic Overview.....	25

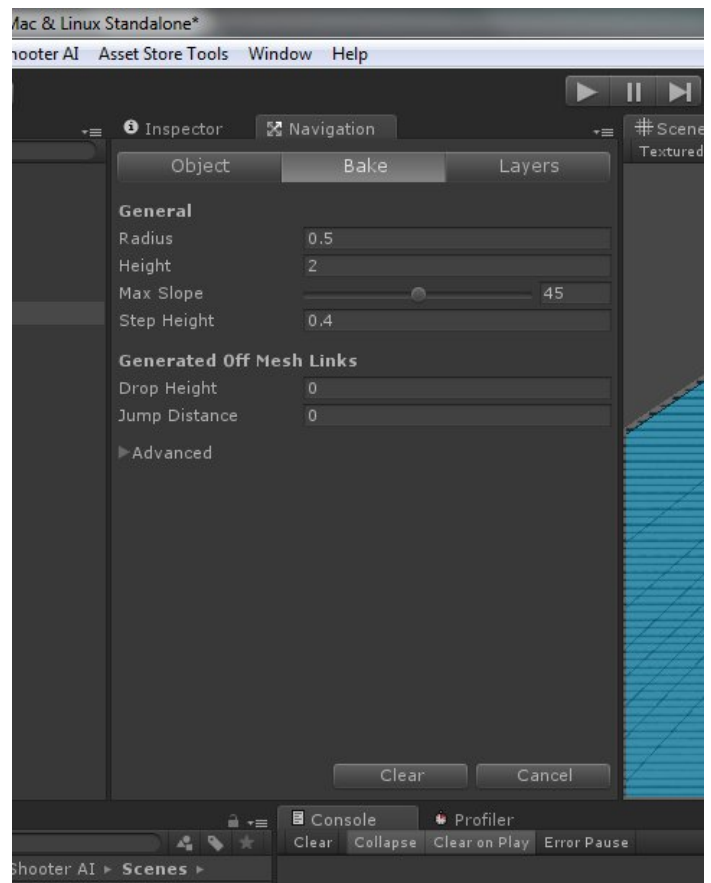
Quick Setup

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. All the tutorials use () to show what to do, and just **bold** instructions to click the specified 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) Bullet setup

First we shall setup the bullet that the gun will be using. In reality its a Gameobject with the bullet tag, a mesh renderer, a deactivated collider, BulletHealthExtras and BulletAutoDestroy scripts and a rigidbody, though we have provided some scripts so that it cleans up the bullets and copes with a PhysX bug. We recommend you use the bullet prefab we provided under **“ShooterAI/Prefabs/Bullets/Bullet1.prefab”**, and then modify it for your own usage.

II) Empty magazine setup

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.

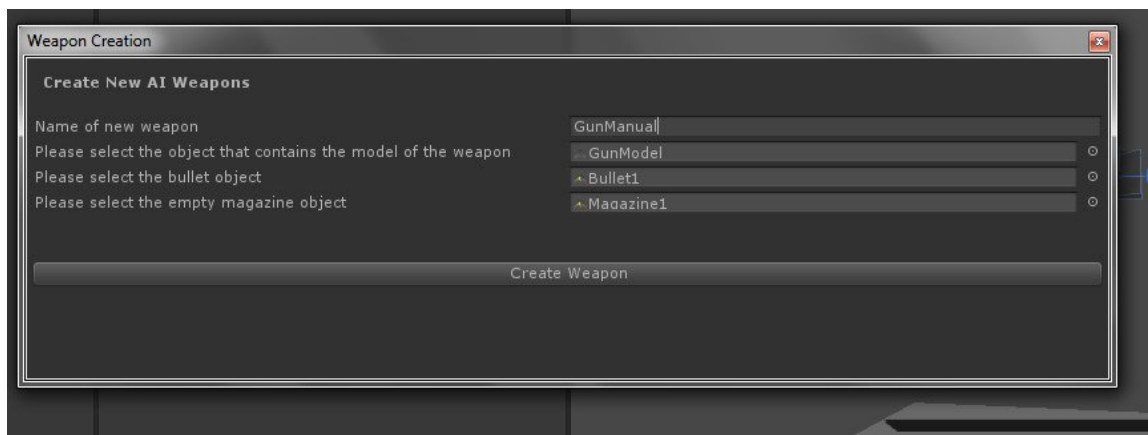
III) Weapon Model

For the weapon model you can use any Gameobject. There can **not** be any type of collider attached to it

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) -> (Drag and drop in the bullet object) -> (Drag and drop empty magazine) -> Create Weapon

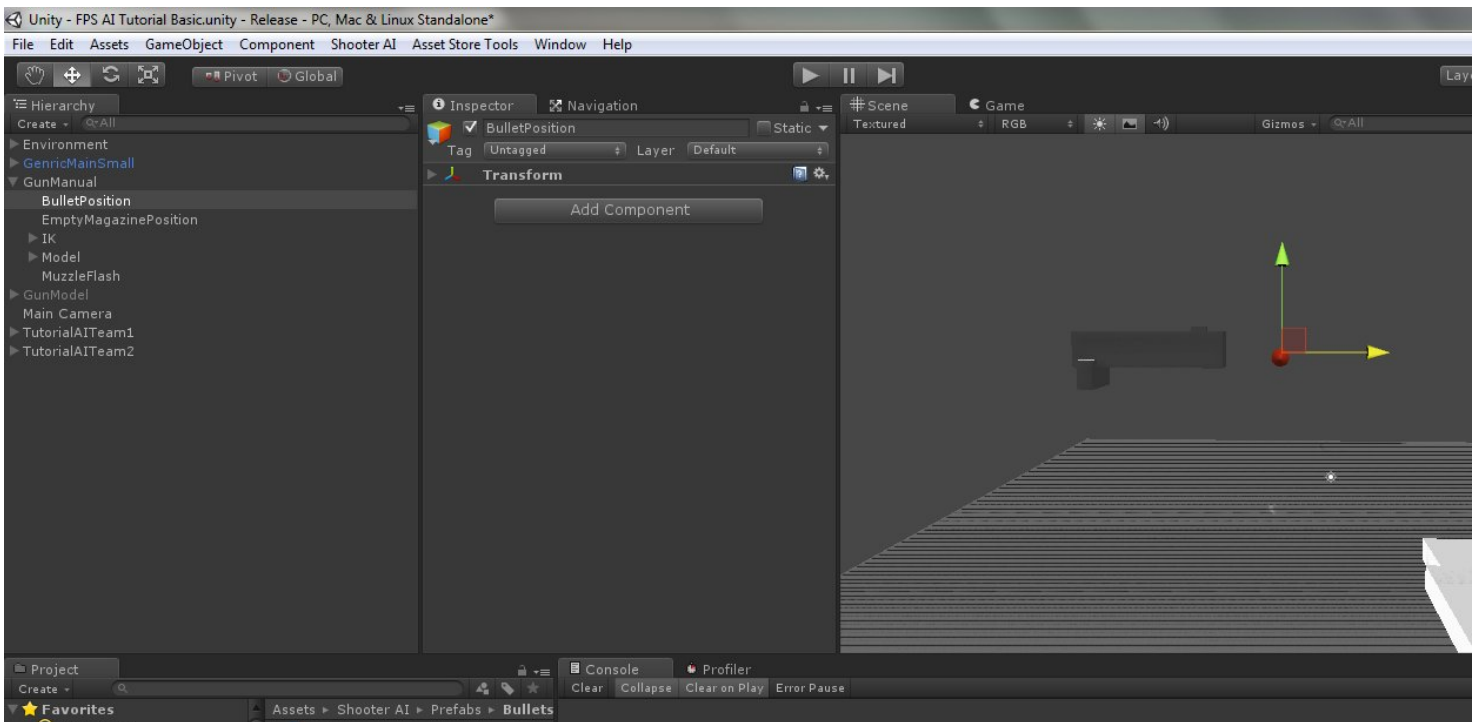


Now your weapon is almost fully setup and we just need to adjust some values.

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.**

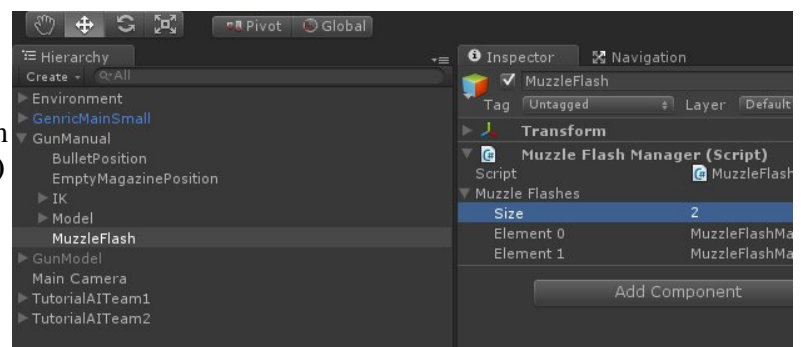
You can overview all parameters by **(Click on the gun object) -> (Open script “AIWeapon Controller Child”)**. The parameters are grouped logically in subcategories. The first group is the “Main Referencing Objects”, which are the objects on which the scripts rely to create bullets, manage IK etc, and is labelled accordingly. You will extremely rarely need to even open this subcategory, as it is managed fully automatically. The next subcategory is the “Ammo Properties”. Finally the last subcategory is the “Shooting Properties” which determine recoil and how much force is applied to the bullet.

Firstly we will configure the bullet creation position. To do this **(Click on the gun object) -> (Open the heirachy) -> (Click on “BulletPosition” object) -> (Position the object to where you want the bullets 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 “MuzzleFlashManager” script) -> (Open array called “Muzzle Flashes”) -> (State the amount of flashes) -> (Drag and drop the flash prefabs into the slots)**



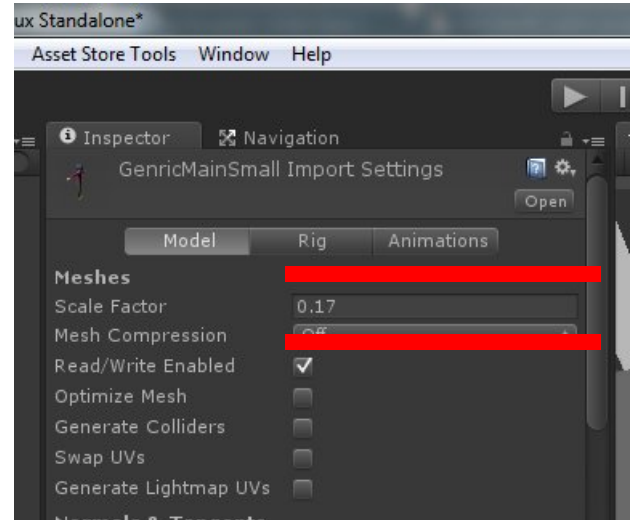
3. AI Character Setup

Preparations

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

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!!!!**

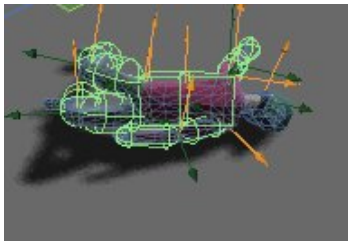


Once we're happy with the model and its properties, **drag and drop the model into the scene.** After that create a ragdoll object out of it. Here are some good tutorials, or look into Unity's documentation on it:

https://www.youtube.com/watch?v=dCwNaE_eVsM

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

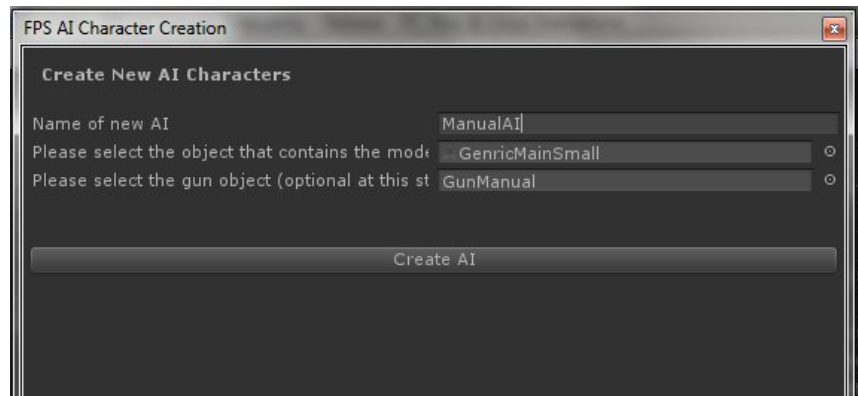
Once you've setup the ragdoll, test it in play mode and see if it tumbles as wanted.



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

Character Setup

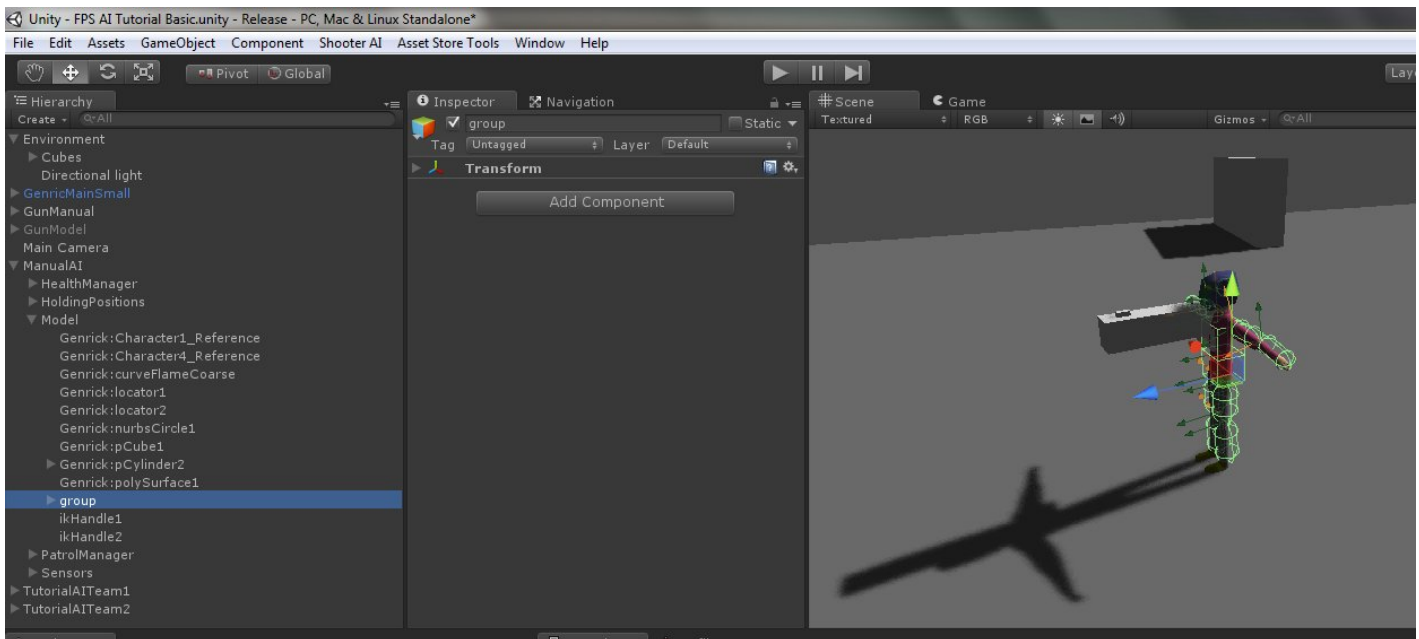
Shooter AI -> Create New Character -> (Type in name for new chacter) -> (Drag and drop in the object with the model+ragdoll) -> (Drag and drop in the gun object we made earlier) -> Create AI



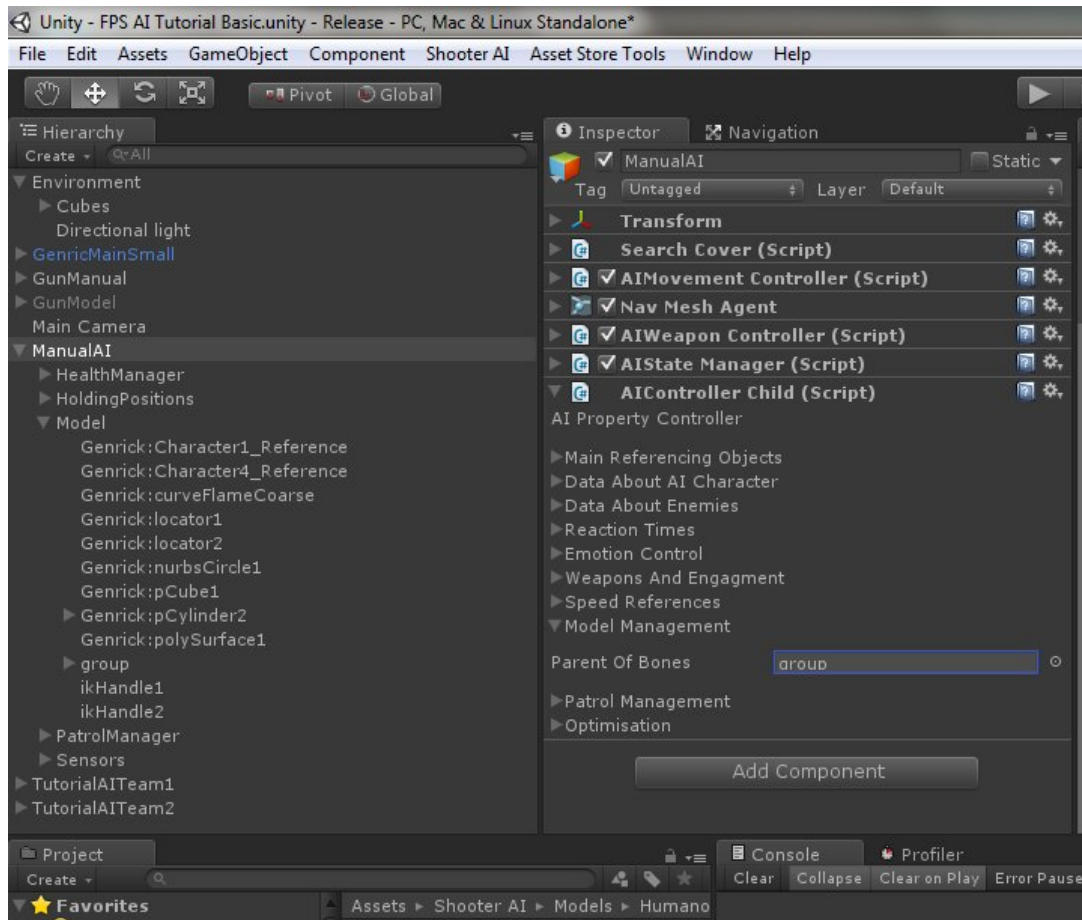
We're almost finished.

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 "AIController Child")**, and then you can view any property that influences the AI.

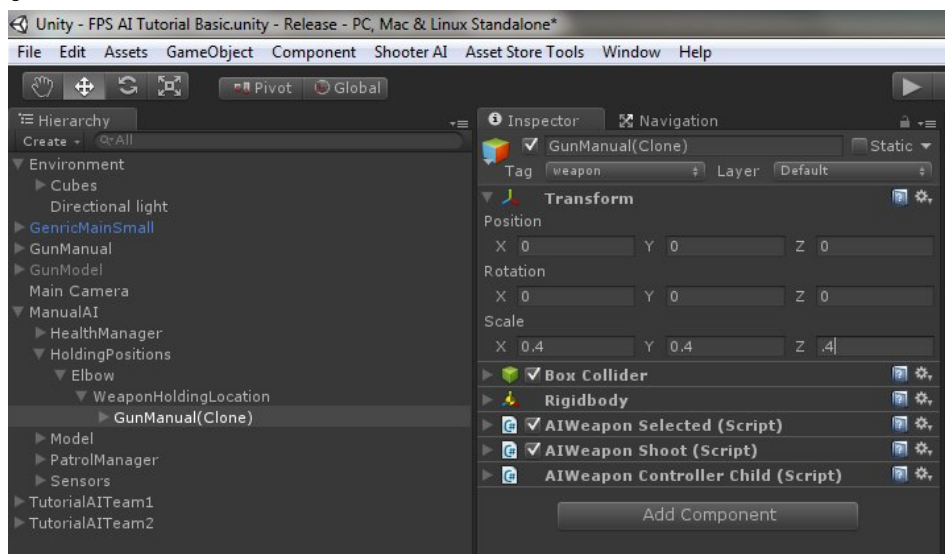
Due to each model being made differently, Shooter AI can not always determine the ragdoll object correctly, and must sometimes be hand designated. To do this, **(Click on the AI Character object) -> (Open it's hierarchy) -> (Click on "Model" object) -> (Open the hierarchy of the "Model" object) -> (Find object that highlights ragdoll when selected, see below)**



(Once you found this object, remember it, and click back on the main parent without closing the hierarchy) -> (Open “AIController Child” script) -> (Open “Model Management” subcategory) -> (Drag and drop the object that we previously found into the “Parent of bones” slot, see below).



You may want to do some modifications to the gun e.g. Scaling. For that (Click on main character) -> (Open its hierarchy) -> (Open “Holding Position” hierarchy) -> (Open “Elbow” hierarchy) -> (Open “WeaponHoldingLocation” hierarchy) -> (Click on gun object)

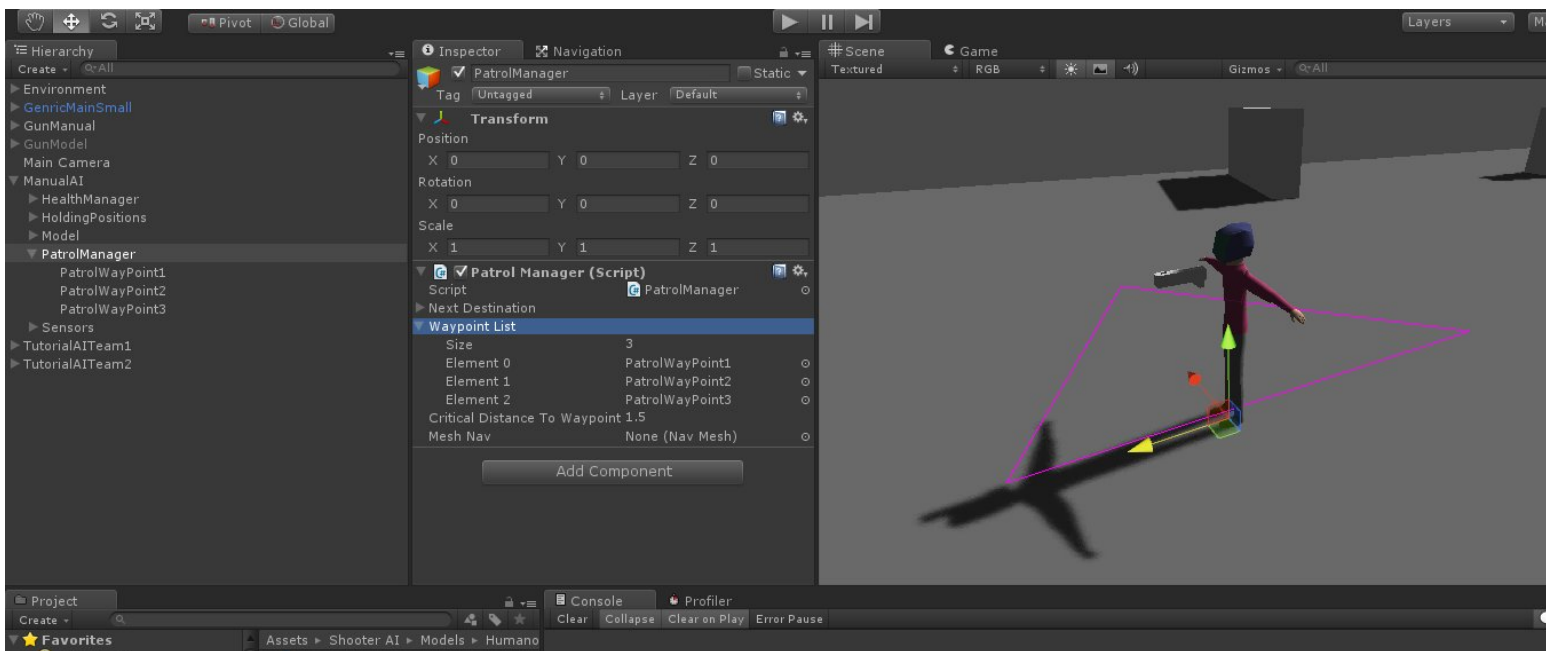


Finally we'll setup our waypoints and then we will be finished!

(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.



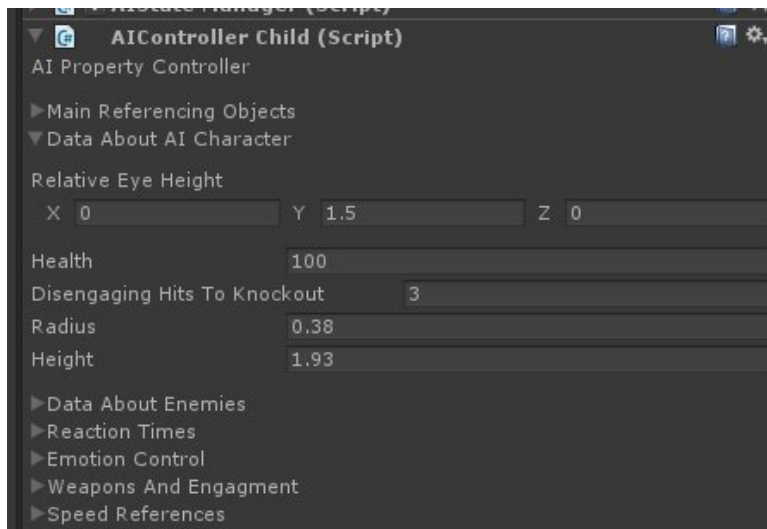
You have finished! After a practising it a couple of times, it is possible to make full AI in under 5 minutes! Have fun!

Also watch our YouTube tutorials on more visual documentation.

Detailed Property Overview

Player properties

Main Properties



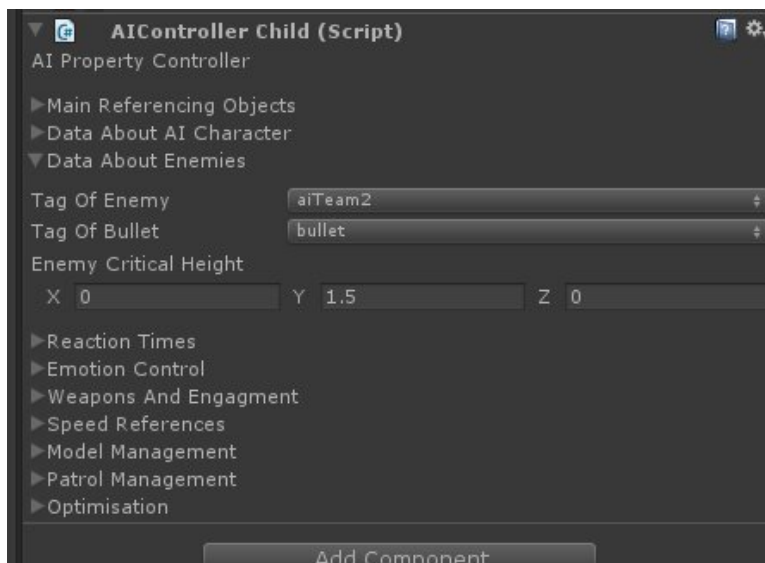
Relative eye height – This is the height at which the eyes are located

Health – This is the amount of health. Once this value is 0 or less, the character dies

Disengaging Hits To Knockout – How many hits to designated “disengaging areas” make the character go disengaged (like death, but in theory still alive, good for stealth games where you need to disengage enemy instead of kill)

Radius – The radius of the NavmeshAgent (character)

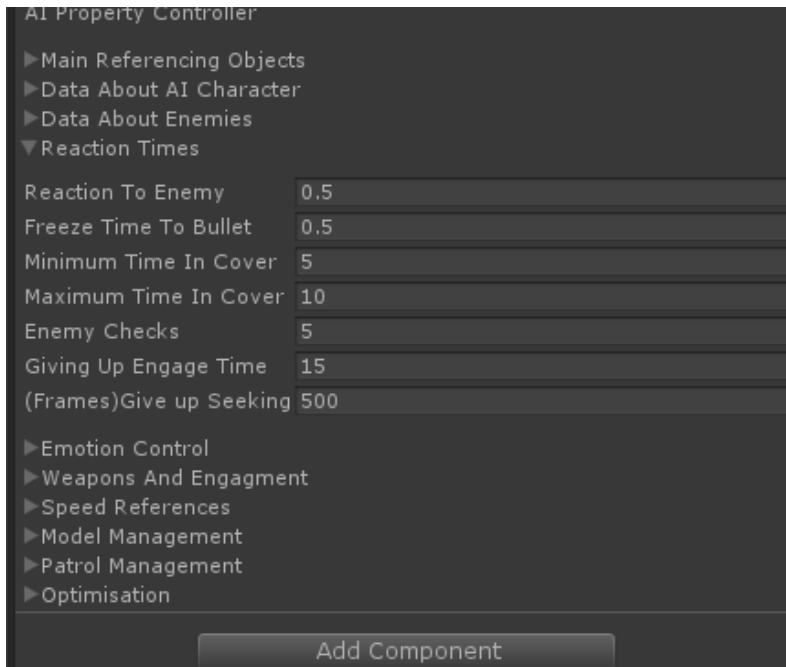
Height – The height of the NavmeshAgent (character)



Tag of enemy – The tag that all enemies have

Tag of bullet – The tag that all bullets have

Enemy Critical Height – The relative position to the ground of where to aim for a critical hit in an enemy (e.g. chest area in human charcters)



NOTE: ALL THESE VALUES GET INFLUENCED BY FEAR AND ADRENALINE!

Reaction to enemy – How many seconds the enemy will freeze when the AI suddenly sees the enemy

Freeze time to bullet – How many seconds of shock the AI will have when it suddenly hears a bullet

Minimum time in Cover – Once the AI is in cover, what is the minimum amount of seconds it will stay there

Maximum time in cover – Once the AI is in cover, what is the maximum amount of seconds it will stay there

Enemy checks – When the AI sees a potential enemy, it will run an “engagement” script that will decide whether to attack or step down. This value changes the minimum amount of seconds between such checks

Giving up engage time – When the AI is in engagement mode and loses sight/hear of the enemy, after how many seconds it should go into engagement mode

(Frames) Give up seeking - If we see an enemy and then lose him again, how many seconds should we investigate before going back to patrolling.

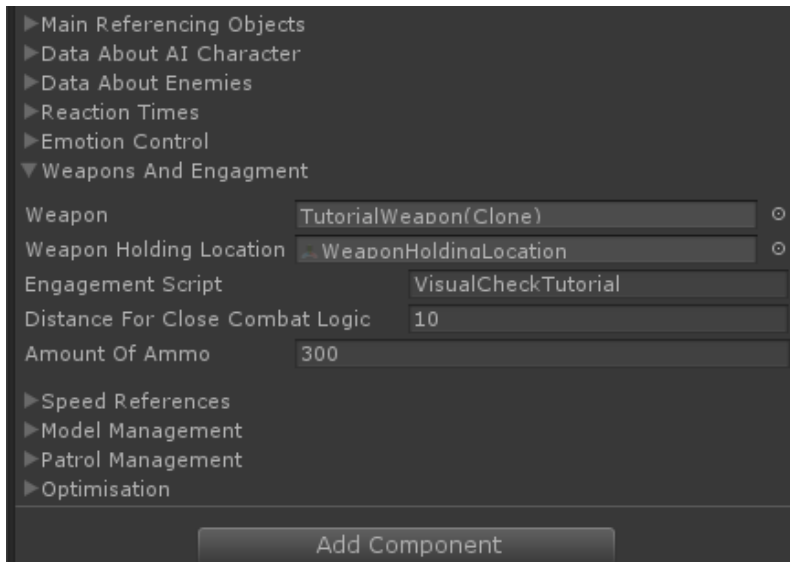


NOTE: THESE VALUES CHANGE VERY OFTEN

Adrenaline - The amount of adrenaline in the system. Adrenaline usually increases the AI character performance

Fear – The amount of fear that the AI character has. Fear is good in small amounts, after which it suddenly turns to panic

Chance For Fight – The percentage to fight when the “Flight or fight” instinct arises. This value is based on fear and adrenaline



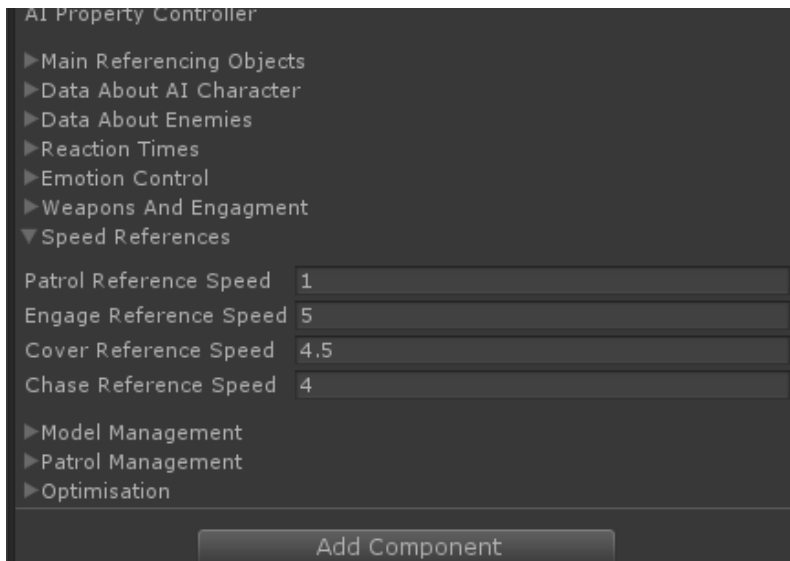
Weapon – This is the object that contains the gun. This object should be created using the Shooter AI gun creation wizard

Weapon Holding Location – This is the object that holds the gun (like the hand), that should be linked to an “elbow” object, for more realistic gun rotation

Engagement Script – This script gets activated once the AI sees the enemy. There is extra documentation on how to write this script

Distance for close combat logic – The distance after which close combat logic will be activated (the AI will try to charge more than hide)

Amount of Ammo – The amount of ammo that the AI has



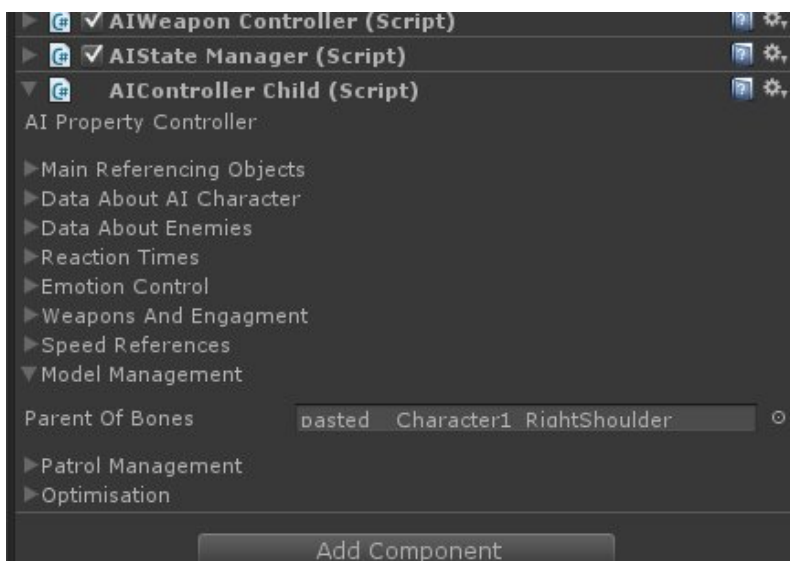
NOTE: THESE VALUES ARE HEAVILY INFLUENCED BY FEAR AND ADRENALINE

Patrol Reference Speed – The speed at which to patrol

Engage Reference Speed – At which speed to run when engaged in a fight

Cover Reference Speed – At which speed to run to cover

Chase Reference Speed – At which speed to chase a fleeing enemy

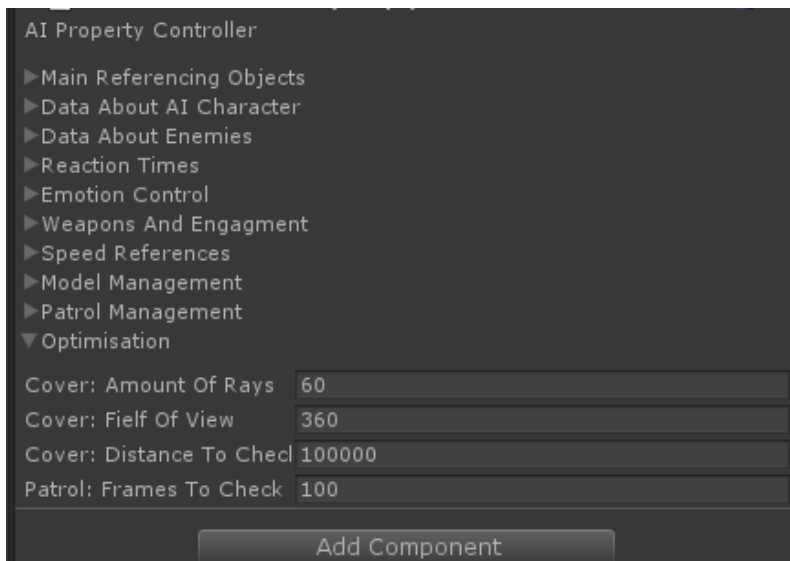


Parent of bones – This is object inside the Model’s hierarchy that contains the ragdoll and the bones of the model



NOTE: THE WAYPOINTS THEMSELVES NEED TO BE MANAGED INSIDE THE PATROL MANAGER

Distance to register waypoint – This is the distance to register a waypoint, so if the distance from the AI character to the waypoint is less than this value, it will count as if the waypoint has been passed and will go to the next waypoint.



Cover: Amount of rays – This is value states how many rays will be cast to figure out the best cover position. The more rays, the better the cover position, but higher performance cost. The less rays, the worse the cover position but less performance cost

Cover: Field of View – The field of view from which to send rays (360 gives most realistic results)

Cover: Distance to Check – The length of the rays that check the best cover position

Patrol: Frames to check – This is how often to check the patrol status. This value makes virtually no difference >50

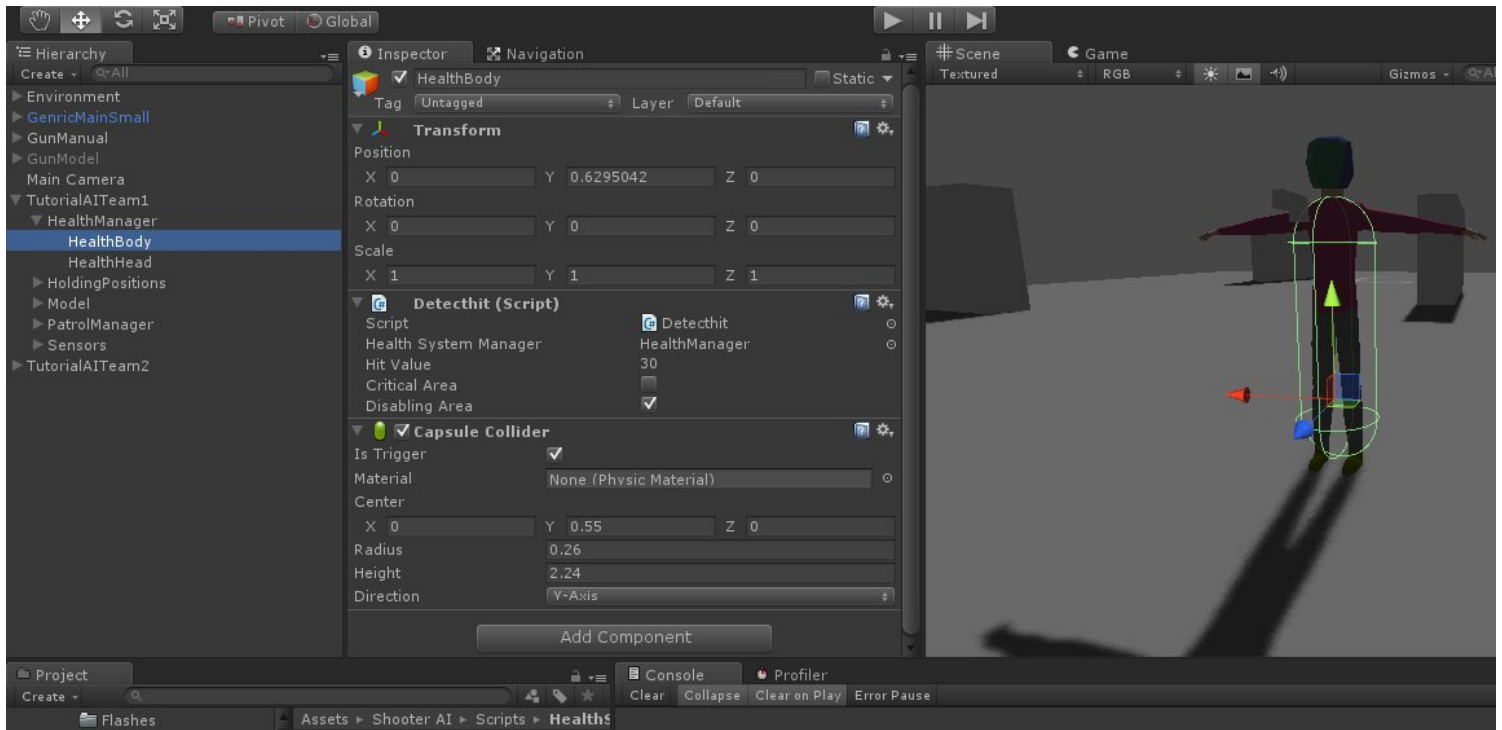
Health Engine

This part describes the health engine and how to use it. The health engine is modular and uses region specific tactile input to register impacts and health reduction. Also it is possible to mark regions as “disengaging” meaning that hits to this region will disengage (not kill) the AI after a specified number of hits.

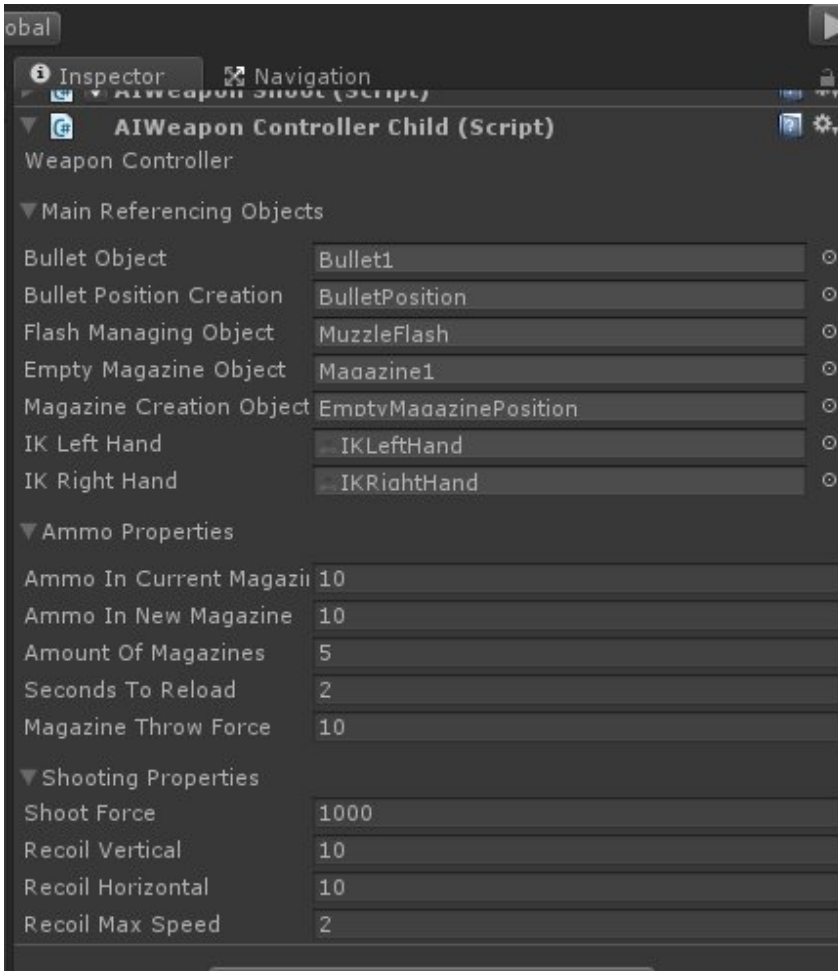
Each region is a different gameobject, preferably a child of the HealthManager. Also each region needs to have a collider that represents that region and **has to be marked as a trigger**. The script “DetectHit” needs to be added:

- **Health System Manager** needs to be set to the HealthManager
- **Hit Value** is the amount of HP that get deducted if a bullet hits this region
- **Critical Area** marks whether the AI character will die instantly if this area is hit
- **Disabling Area** marks whether this region will contribute to “disengaging” hits, meaning that after a number of hits to “disengaging” areas, the AI Character will become disengaged which means that the character is unconscious (it is possible to modify this through the HealthManager’s “AIHealth Executer” script; see comments inside)

By default, each AI Character is supplied with 2 impact regions. Below you see the “body” region.



Gun Properties



Main referencing objects: This subcategory is rarely used and is managed fully automatically. Only experienced users should modify this.

Ammo Properties:

Ammo in current magazine – The amount of ammo in the magazine which is in the gun

Ammo in new magazine – How much ammo should be in each new magazine

Amount of magazines – How many magazines are present, though this number gets changed automatically by the AI character that holds it

Seconds to reload – The amount of seconds it takes to reload a new magazine

Magazine Throw force – With which force is the empty magazine going to fly out. This is only for visuals

Shooting Properties:

Shoot force – The amount of force that is applied to the bullet when it is created.

Snipers have larger number whilst close range weapons have smaller values

Recoil vertical – The amount of vertical recoil

Recoil horizontal – The amount of horizontal recoil

Recoil Max Speed – The max speed at which to jolt the gun to the recoil values

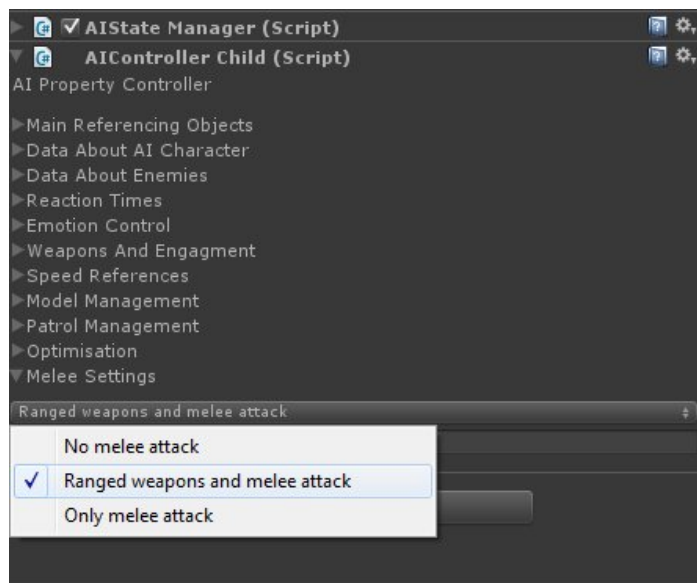
Melee setup

Melee attacks are extremely easy to setup and you make animations directly inside of Unity! AI characters can use a mix of fighting techniques meaning the AI will use its gun as a ranged weapon first, but then as a melee weapon at close ranges. Also, if you have Unity Pro, you can animate IK together with any melee animation, making epic fights and swings. But as usual in Shooter AI, we provide basic animations for out of the box functionality.

After you have created a weapon, click on it and open the **Animation** component -> **Animations** -> (Resize accordingly and throw in your animations)

AI Character Setup

It is extremely easy to setup melee for AI characters.

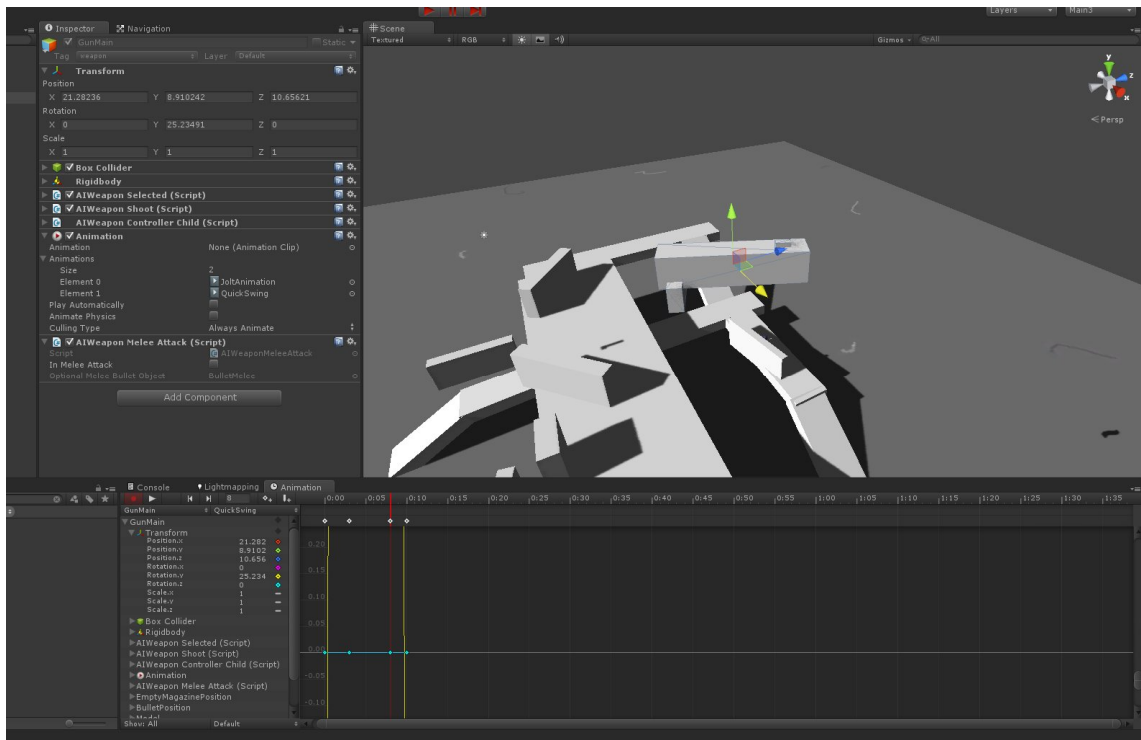


(Open your AI character) -> AIControllerChild -> Melee Settings -> (Select the type of melee attack you want)

The distance for melee attack sets the distance after which to attack. If the AI isn't attacking from far enough, increase this value.

Melee Animations

To make melee animations first **Window -> Animation -> Create New Clip -> (Save your animation) -> (Click on the record button) -> (Record your animation; we recommend watching tutorials on YouTube for better explanations)**



After that your finished! The animation is inside your weapon by default!

How To Write Own Engagement Script

To fully use Shooter AI's potential, you should use engagement scripts to customize when the AI attacks or steps down.

Engagement scripts are scripts that receive a message once they are created, decides using code what to do, then sends back a message telling whether to attack or step down. This lets you create AI that only attacks under specific circumstances.

Upon creation the code receives a message called "VisualCheck" will be sent to the script. Then you can write your own code on what to do (see example below) and then send back message either "Engage" or "StepDown".


Below we see an example where a random number between 0 and 100 is picked, and then if the number is larger than 90 then the AI character will engage the enemy. Effectively this means that there is a 10% chance of the AI character engaging the enemy upon visual contact.





```
1 using UnityEngine;
2 using System.Collections;
3
4 public class VisualCheckTutorial : MonoBehaviour {
5
6
7 void VisualCheck()
8 {
9
10
11 float randomNumber = Random.Range(0f, 100f);
12
13 if(randomNumber > 90f)
14 {
15     SendMessage("Engage");
16     Debug.Log("Engage");
17 }
18 else
19 {
20     SendMessage("StepDown");
21     Debug.Log("Stepped Down");
22 }
23
24
25 Destroy(this);
26 }
27
28
29 }
30
```

Ready

- ▶ Main Referencing Objects
- ▶ Data About AI Character
- ▶ Data About Enemies
- ▶ Reaction Times
- ▶ Emotion Control
- ▼ Weapons And Engagment

Weapon 

Weapon Holding Location  

Engagement Script

Distance For Close Combat Logic

Amount Of Ammo

- ▶ Speed References
- ▶ Model Management
- ▶ Patrol Management
- ▶ Optimisation

Add Component

Ultimate FPS with Shooter AI

Just add in the UFPS player prefab, and your done!

Team Control

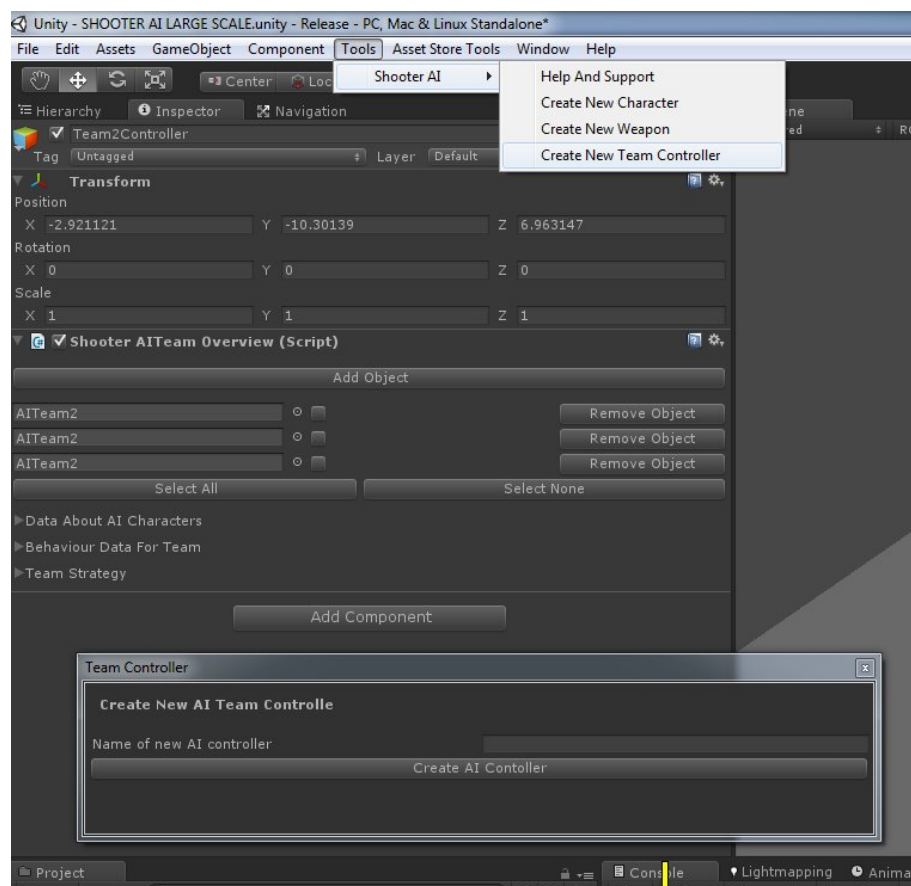
In Shooter AI it is possible to create dynamic teams with leaders that follow epic strategies to win against their enemies. Also you can edit almost all values of any/all team members from the team control panel.

Team Creation and Management

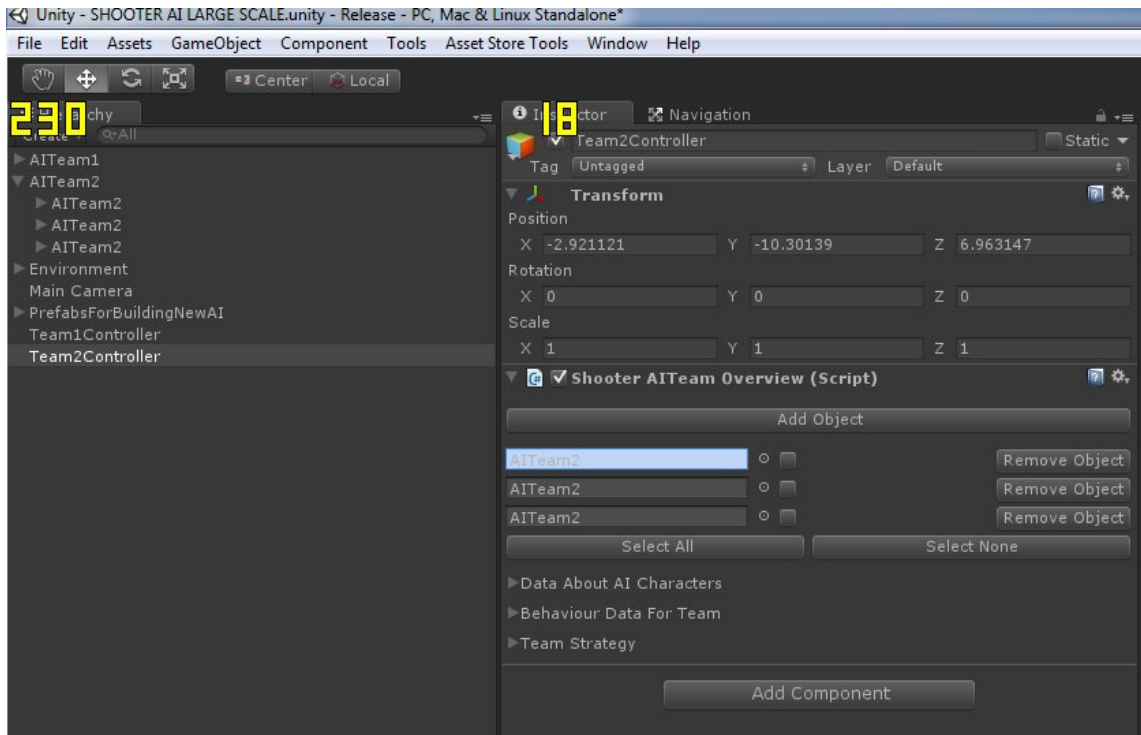
Team Creation

It is extremely easy to create and manage teams with Shooter AI.

To create a team, **Tools -> Shooter AI -> Create New Team Controller -> (Fill in name for the controller) -> Create AI Controller**



Next we add the team-mates. To do this, first **Add Object -> (Drag and drop the team mate to that object)**



By default, the first AI character on the list is the captain (highlighted green), but this changes dynamically when the characters start dying in game. We recommend using the most powerful character as the team captain, as this will get more fun results.

Team management

You can manage the team easily from the team controller panel extremely easily.

To manage AI character properties, **(Select the objects you want to apply to by ticking the check box to the respective name) -> Data About AI Characters -> (Change data that you want)**

WARNING:

- Only the selected objects will be edited
- The changes are applied instantly
- The hand wielding gun always gets changed if you open the "Model" tab

Behaviour and Event Management

You can make scripts apply at vital events by **Behaviour Data For Team -> (Write in the name of your script and the relevant event name)**

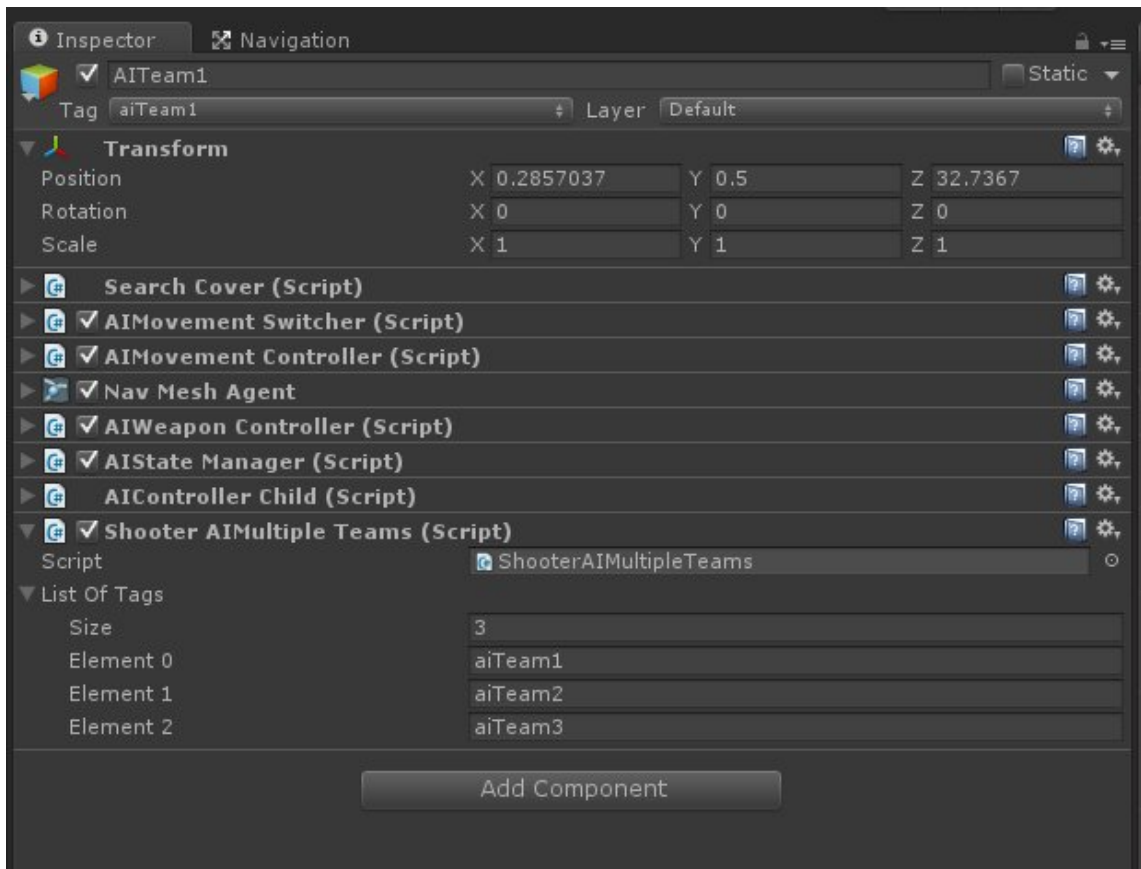
Team Strategy

To make the team follow a specific strategy click on **Team Strategy** -> (**Select a predefined strategy or select "Custom" and specify your strategy script's name**)

You can base your strategy script off the "ShooterAIStrategyCloseTogether" script. The API can be located in the "ShooterAITeamOverview" script. Everything is commented and self-explanatory.

Multiple Teams

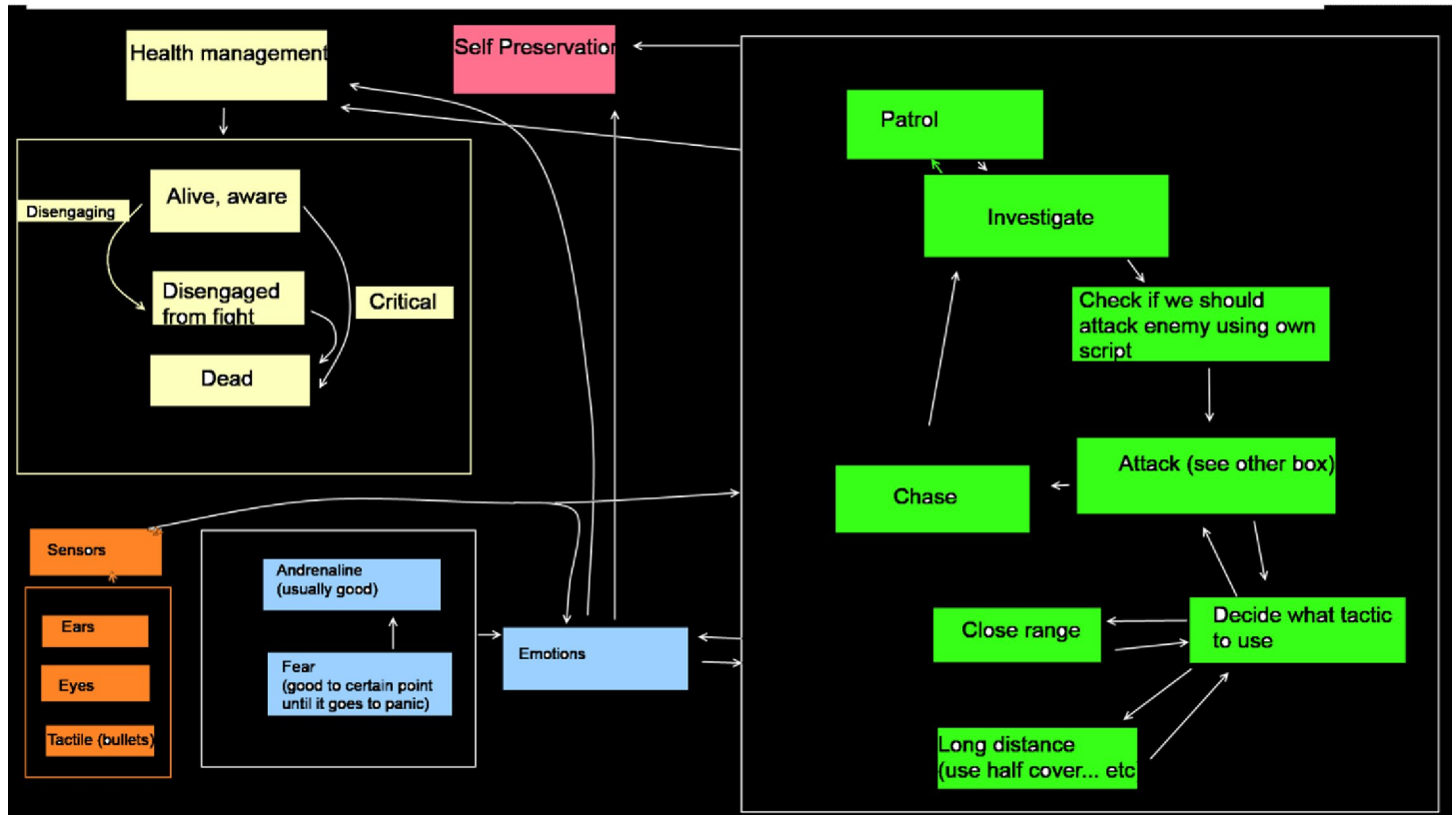
Using a small script you can let the AI fight as many teams as you want. To do this
(Click on your AI character) -> **Add Component** -> **Shooter AI** -> **Multiple Teams** -> (Open "List Of Tags") -> (Make the size the amount of teams you want the AI to fight) -> (For each element in the list, enter the tag of an enemy team)



The algorithm uses distance to find the closest team. This isn't always the best and we highly recommend you to modify this script.

Logic Overview

Here is the main overview of the Shooter AI.



Main cycle: The AI starts in Patrol. If it suddenly hears/sees the enemy it will go into Investigate, where it will try to determine where the enemy went and find him. If the AI sees the enemy, it will run a script (Set in the Inspector) that will return whether to engage or to step down. After that, if the AI steps down, it will go into Patrol mode, or else it will start executing the shooting logic. There are 2 types of fighting logic, close range and long range, which are chosen by the parameter ("stop walking distance" in AIStateManager). The close range makes the AI storm the other player and directly shoot. The long range logic makes the AI try and find cover, and then attack from there. It peeks out, shoots then goes back to cover. If it still hears bullets, it stays a random number of seconds, before going out and shooting again. If the AI is hit, it will determine whether to fight or flight. If it fights, it will storm try and storm the enemy, else go into cover. If the AI loses the enemy, it will go into Investigate/Chase, seen by how the enemy was going. Then, if the enemy is gone (eg. dead), the AI goes back to Patrol.

Emotions: The emotion component has 2 parts: adrenaline and fear. Adrenaline is determined by engagement, surprises (eg. suddenly seeing enemy). Fear is determined by getting hit, initial fear and hearing bullets. Adrenaline makes reactions quicker, increases FOV, increases hearing distance and some other minor factors. Fear first makes adrenaline, then after a certain point starts going slowly into panic and after a certain point, goes into full panic and starts shooting everywhere.

Sensors: The AI uses 3 sensors to determine the world around them: Eyes, Ears, and tactile. The Eyes and Ears are children of the "Sensors" object and can be controlled by individual scripts. Tactile sends only whether it got hit or not, using the "Health Manager" that sends out a message when the AI is hit.

Self Preservation: Self Preservation is a function executed every step (called "SelfPreservation") and overrides a couple of actions, for example go into cover when the AI is in full combat and needs to reload. See the function for exact paramaters.

Health Management: The health management is broken into 2 main parts: the health manager that decides when to die etc, and the receptors that detect bullet hits, what type they are (critical, disengaging, normal) and how much damage they do.

Critical: Instant kill.

Disengaging: After a number of disengaging shots (determined in the Health Manager) the AI will go "disengaged", so not dead, but out of the fight.

Normal: Only take away a number of health points.

All scripts and manual © Copyright Gateway Games 2013

