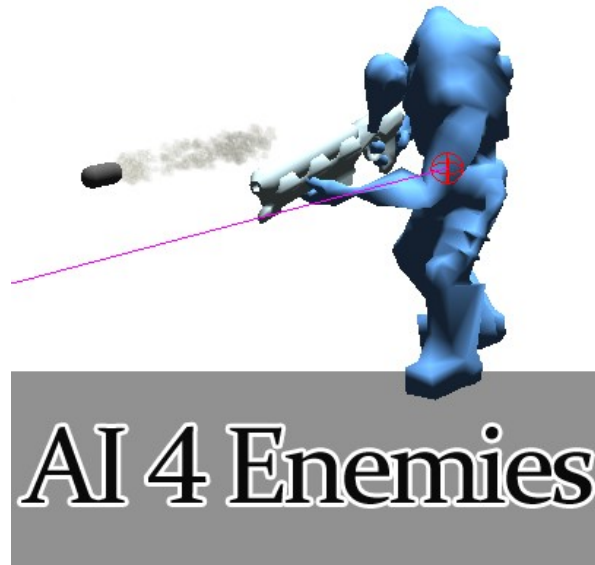


# AI 4 Enemies



Give your NPC models artificial intelligence to create close combat fighters and range weapons-user.

"AI 4 Enemies" supports animations, sounds, different damage modes like damage by calculation or by collision, obstacle avoidance, target following, dynamic waypoints and vertical height. You can create combat fighter and range weapons-user. It includes a waypoint editor for complex patrol routes and it supports `c#` events for a better integration into your.

## Scripts and objects

### AI Behaviour

This component give your character model artificial intelligence. It consists of the following sub components which I'll explain in the next chapters:

- AI Animations
- AI Controller
- a waypoint editor,
- a target point,
- a rigidbody (if your object hasn't one)

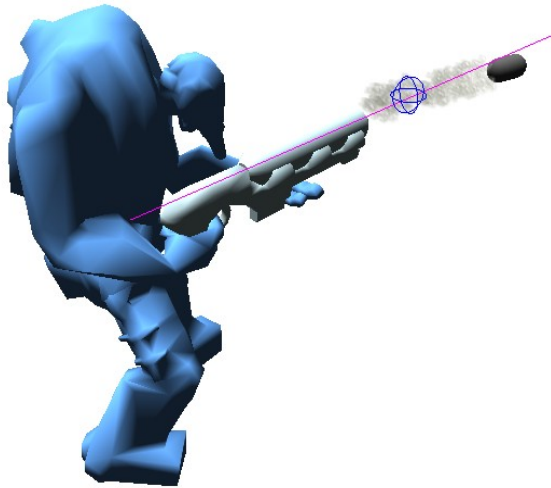
### Scripts

In this sub menu you can find additional components which can be used to add more functionalities to AI characters.

### Spawn Point

That's the point where shooting projectiles of AI characters have to be initialized.

*Hint: If you use calculation damage it can be used for special effects while fighting.*



Blue sphere: Spawn Point

### Collision Damage Controller

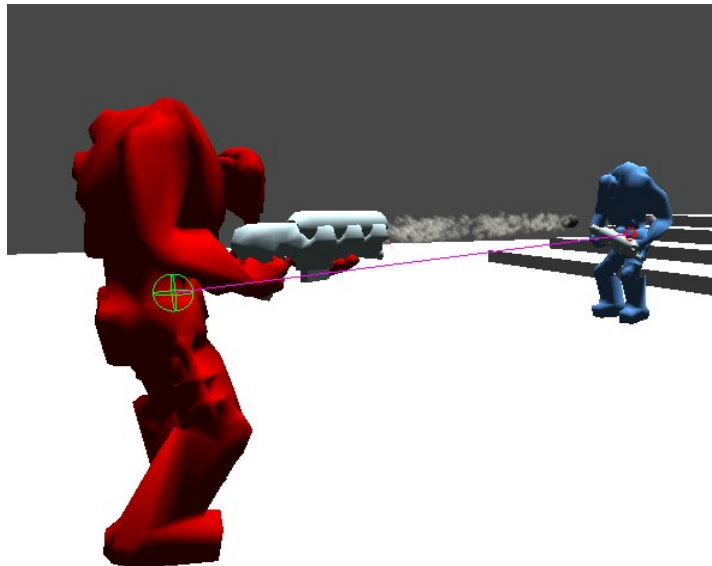
If the ai character has to receive collision damage, which damage value has to be calculated by the character itself, add this script to the ai character.

*Hint: For using this functionality, your character need a collider component e.g. a capsule collider.*

### Target Point

This is the key object for detecting other objects and targets. Both of ai characters and target objects must have a target point.

*Hint: For a good game play it's utile all characters have there Target Points on a similar level.*



Green sphere: Target Point of AI character

Red sphere: Target Point of player character

## AI Controller - parameters

### Target

The target object to attack.

### Target Point Height

The y position of the target point.

### Spawn Point

If available it's the slot for the point where the projectile has to be initialize

### Projectile

If available it's the slot for the prefab which has to initialize by shooting/punching

### Walk Speed

The speed how fast the character walks.

### Run Speed

The speed how fast the character runs to his target when the AI discovers it.

### Rotation Speed

The speed how fast the character rotates.

### Field of View

The field of view where the target can be detected

### View Distance

The distance where the target can be detected

### Attacking Distance

The distance where the npc begins to attack (shooting/punching)

### Minimal Distance

The character stops at this distance. e.g. ranged weapon user has 10, for close combat it has 1.

### Weapon Range

The range of the weapon (measured from character center).

### CrossFade Animation

Use „CrossFade“ for starting animations. By default parameter is disabled, so „Play“ will be used.

### Shooting Pause

The pause between two shots or punches.

### Death Delay Time

The time in seconds how long the character appears after it will be killed. 0 means it will be destroyed immediately. A negative number (-1) means no destroying.

### Projectile Force

If using projectiles, this is the speed of the projectiles.

### **Fight Effects Delay**

It's the delay between starting the fight animation and playing fighting sound and initializing the projectile prefab.

### **Calculation Damage**

For applying damage AI tries to call the ApplyDamage function of the target.

### **Min Damage**

When using calculation damage this is the minimal damage.

### **Max Damage**

When using calculation damage this is the maximum damage.

### **Hit Chance**

When using calculation damage this is the chance to hit.

### **Health**

The health of the character

### **Play Sound**

Play sound by actions

### **Discover Target Sounds**

Sound files and their volumes which have to play after AI discovers its target (played by random).

### **Fight Sounds**

Sound files and their volumes which have to play during fighting animation (played by random).

### **Hit Sounds**

Sound files and their volumes which have to play during hit animation (played by random).

### **Die Sounds**

Sound files and their volumes which have to play during die animation (played by random).

### **Idle Start**

Start character by standing. If it's deactivate the character starts in waypoint modus and goes patrol.

### **Waiting Max Time**

Maximum time for waiting on a waypoint during waypoint modus.

### **Visible Layers**

Which Layers has to be recognized by walking and which has to be ignored.

*Hint: If you use projectiles, remove the layer of them from this list.*

### **Waypoints**

If you don't activate „Idle Start“ the character will patrol these waypoints until he find his target. For creating waypoints use the AI Enemy Waypoint Editor.

*Hint: If you don't create waypoints the script creates two waypoints automaticly*

## **Additional Parameters:**

### **Gravity Factor**

The speed to fall down

### **Waypoint Distance**

The distance from the character where the waypoints will be created automaticly.

### **Obstacle Avoidance Distance**

The distance for obstacles Avoidance

## AI Controller - Events

"AI 4 Enemies" supports C# events which will fire in following situations. The included demo scene shows an example to work with these events.

All events has an `AIEventArgs`-Parameter. It gives the following information about the firing object: name, position, rotation, current health, tag.

### **onHit(AIEventArgs e)**

This Event will be fired when the AI character will be hit.

### **onIsDying(AIEventArgs e)**

This Event will be fired when the a AI character gets the final damage and the die animation starts.

### **onIsDead(AIEventArgs e)**

This Event will be fired when the a AI character will be destroyed.

## AI Controller - Functions

The following functions are public.

### **ApplyDamage(float damage)**

Apply damage to the character. Damage will be substracted from health

### **AddLife(float life)**

Add life to the character. Life will be added to health.

## AI Animations - Parameters

### **Idle Animation**

The slot for the idle animation

### **Walk Animation**

The slot for the animation which plays when character runs/walks.

### **Fight Animation**

The slot for the animation which plays when character shoots or punches.

### **Hit Animation**

The slot for the animation which plays when character is be hit.

### **Die Animation**

The slot for the animation which plays when character dies.

## **AI Enemy Waypoint Editor - Parameters**

The AI Enemy Waypoint Editor creates the waypoints for this object.

### **WP Parent**

The parent object of the waypoint which will be created.

### **WP Prefix**

This is the prefix of your waypoint names.

## **Collision Damage Controller - Parameters**

### **Size**

Quantity of object tags which can recognize by this controller.

### **Tag**

The tag name of objects which cause this damage.

### **Min Damage**

The minimal damage that caused objects with this tag.

### **Max Damage**

The maximum damage that caused objects with this tag.

## **Spawn Point Behaviour - Parameters**

### **Look At Target Point**

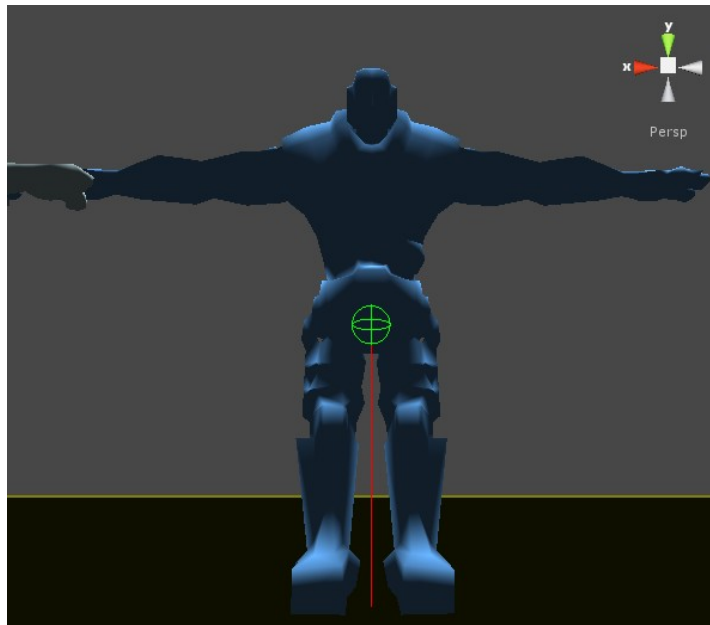
If you work with projectiles, the object fly in the direction of the Spawn Point Rotation. If you activate this parameter, die Spawn Point rotates to your target. This is better for hitting your target.

## Quick Start - Tutorials

In this chapter explains how you can add AI to your character models. You have many opportunities so I limit these examples.

### Minimum AI initializing

- Add „AI Behaviour“ to your enemy character model
- Positioned the target point nearly the center of the character and let the red line ended on the bottom side of the characters feed (see the picture below).
- If the character is very big or very small, you can adjust the length of the red line by changing „Target Point Height“ - Parameter of „AI Controller“
- Assign the animations to the AI Animations-Slots
- Drag target object to the target slot
- Add „Target Point“ to the target model (normaly it's the player character)
- Positioned the target point on a similar height like the enemy character



### Fight modes

For creating a close combat character, you can use following parameters:

- Change minimal Distance to e.g. 1
- Change Attaking Distance to e.g. 1.5

For creating a range weapon-user, you can use following parameters:

- Change minimal Distance to e.g. 5
- Change Attaking Distance to e.g. 20

## Damage

### Receiving damage

There are three alternatives to receive damage.

If the player don't use projectiles the player character (the target object) calls the „ApplyDamage“ function of the character AI directly to subtract health.

This could be done like the following:

```
aiGameObject.SendMessage ("ApplyDamage", 5.0);
```

If the player use projectiles there are two ways to subtract life.

The first option: the projectile prefab must have a script which calls ApplyDamage (like above) when it collides. In addition to that you have to do the following:

- Add a collider to the character AI model (if your character hasn't one)
- Change size of the Collision Damage Controller

The second option: The character AI recognizes the collision, calculates the damage value and applies it to itself. For this you have to do the following:

- Adding a collider to the character AI model (if your character hasn't one)
- Change size of the Collision Damage Controller
- Add „Collision Damage Controller“ to the character
- Define the tag name of game object which causes damage to the character
- Define min damage and max damage

### Causing damage

Causing damage could be done by three ways. For testing there is an example script included in the package named „HealthControllerExample“. Add this to your player character. It includes an OnCollisionEnter function which calls an ApplyDamage function which is included, too.

If your character AI don't use projectiles:

- activate Calculation Damage (it's active by default)
- define Min Damage, Max Damage and Hit Chance
- when target is in front of the AI, it calls the ApplyDamage function to cause damage.

If your character cause collision damage with projectiles, there are two ways:

The first option: the projectile prefab must have a script which calls ApplyDamage when it collides with the target object. So the script has to call the ApplyDamage function in the OnCollisionEnter function like above.

The second way is that the target character itself recognizes the collision and applies damage by collisions to itself.

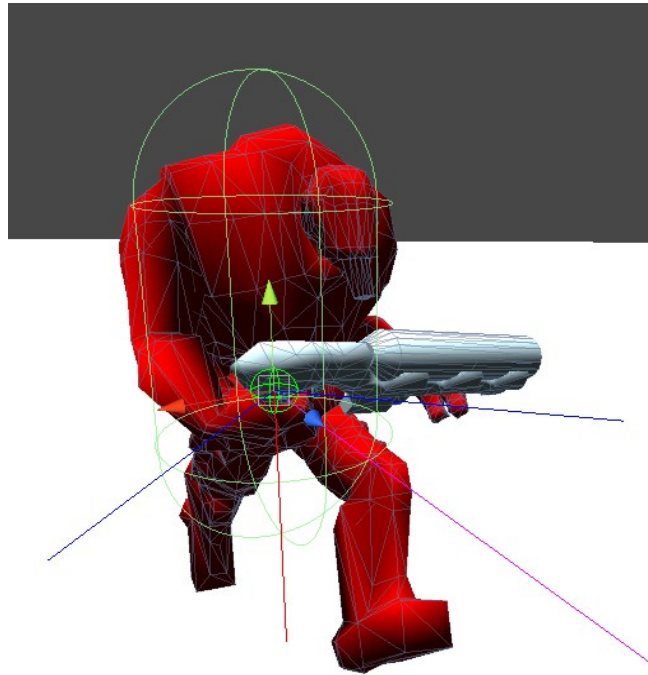
In both cases you have to do the following:

- Deactivate „Calculation Damage“
- Drag a projectile prefab to the projectile slot.

- Adjust „projectile force“ (speed of the projectile)
- Adjust the „Projectile Delay“ to synchronize the fight animation and the initializing of the projectile

Pictures

Picture: all sensors



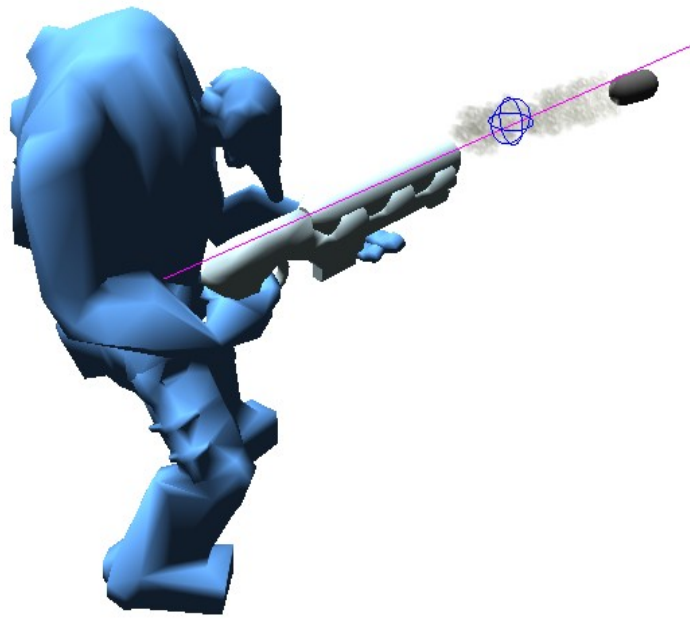
Green sphere: target point of ai character

Red line: target point height

Magenta line: ray to target/next waypoint

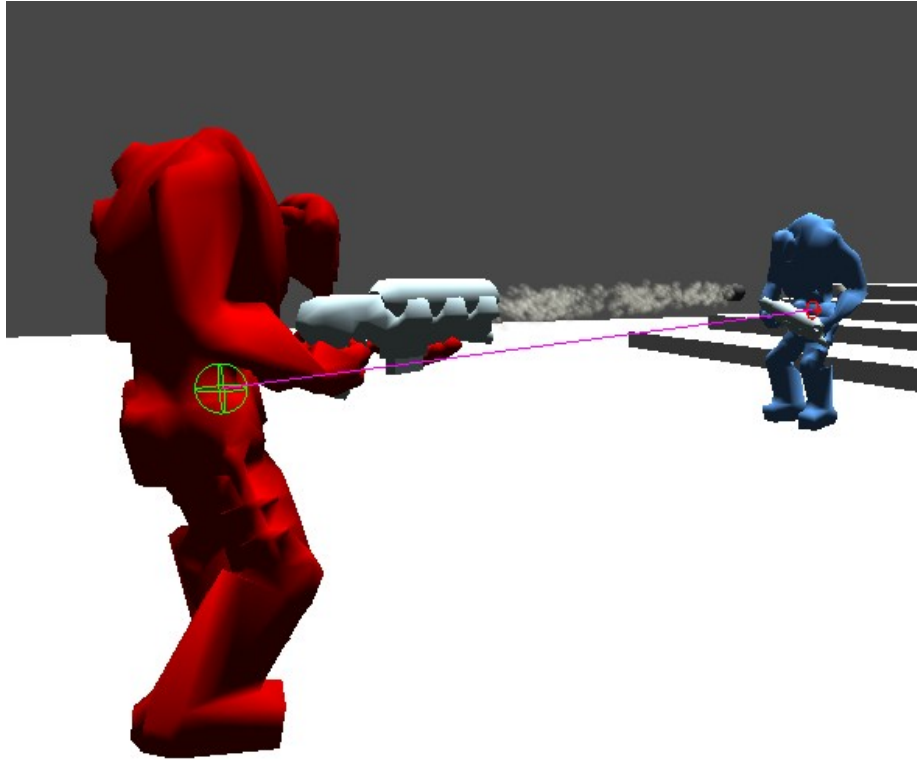
Blue lines: obstacle avoidance distance

Picture: Spawn Point



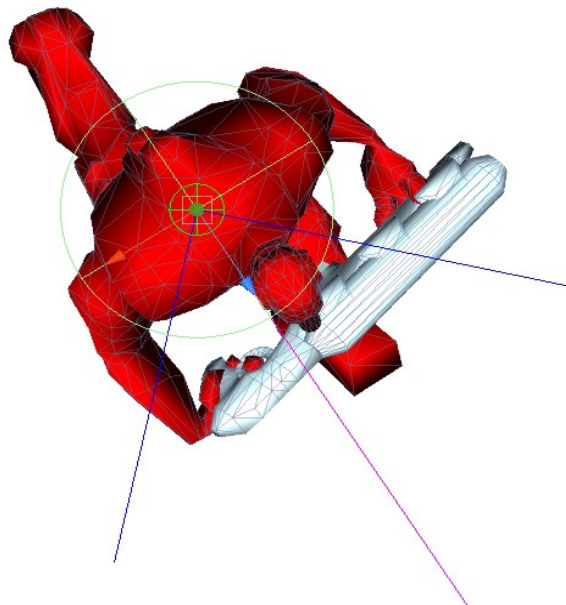
Blue sphere: Spawn Point

Picture: Target Point



Green sphere: Target Point of AI character  
Red sphere: Target Point of player character

Picture: Obstacle avoidance distance



Blue lines: Obstacle avoidance rays for detecting barriers

## Contact

No application is perfect and every tool has bugs. So if you find problems or have some ideas to optimize this tool, you can to contact me at [hummelwalker@seifert-ing.de](mailto:hummelwalker@seifert-ing.de) or Twitter [@hummelwalker](https://twitter.com/hummelwalker).