

Basic AI Controller

User Manual

Contents

Getting Started.....	2
Adding the Controllers.....	2
Using the Controllers	3
Simple AI Controller	3
Basic AI Controller.....	3
Features	6
Character Class.....	6
Character Stats classes.....	6
Levelling Up.....	6
States.....	6
Damage System.....	7
Head Look	8
Editor and SceneView	8
References	9

Getting Started

There are two controllers, a Simple AI Controller and a Basic AI Controller.

Simple AI Controller

It moves in a straight line, can follow targets and destroy itself on collision. It can be used to control bullets, missiles and other things that need automation.

Basic AI Controller

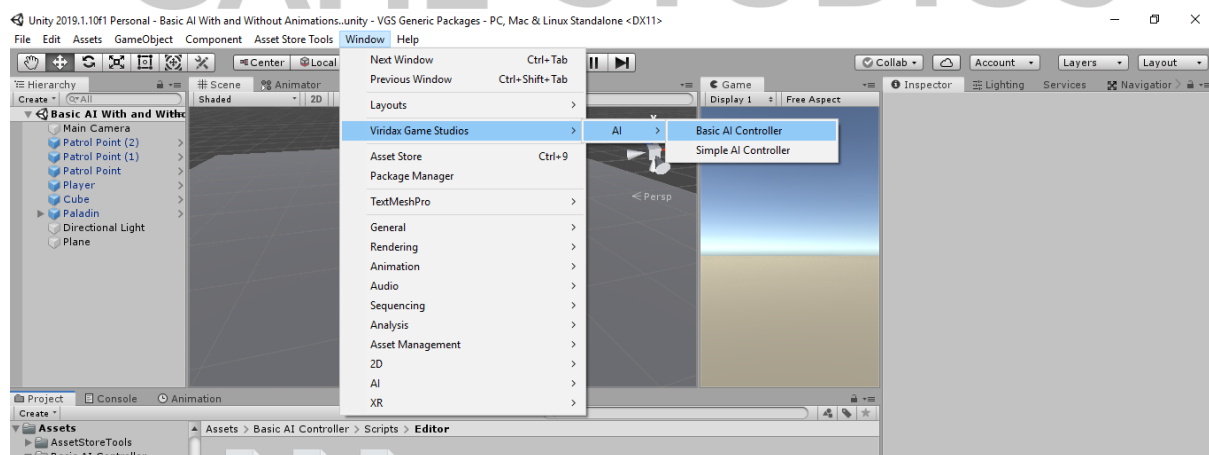
This one is a bit more advanced and supports different states such as patrolling, attacking and more. This can be used to control basic NPC's in games such as spaceships or guards in a castle.

Before you begin using any of the Basic A.I controller script, make sure you have done the following first:

1. Ensure you have animations for Idle, Running (and/or walking), receiving damage and attacking. Idle should be the default animation state.
2. Set the triggers and Boolean conditions accordingly:
 - a. You can use the conditions in the script or use your own.
 - b. If you use your own conditions, you will have to change them in the script as well. (Where it says `anim.SetBool()` and `anim.SetTrigger()`)

Adding the Controllers

In order to add the controller, click on Window -> Viridax Game Studios -> AI -> Basic AI Controller, or Window -> Viridax Game Studios -> AI -> Simple AI Controller.



Alternatively, you can drag the scripts onto any Game Object you want.

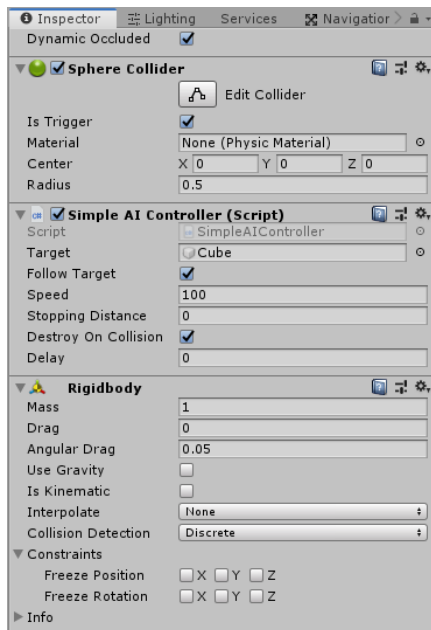
If you added the Basic AI Controller script from the menu, then it will automatically add the “Player” tag to the list of enemy tags. Note: the default reaction is to attack. If you added the script manually, you will have to do this manually from the inspector or in code.

If you need to add more tags, you can do so in the inspector or in the script.

Using the Controllers

Simple AI Controller

This script is straight-forward and easy to use. Once the script has been added to the game object, assign a target and it will automatically move to the target. Any game object can be assigned as a target.

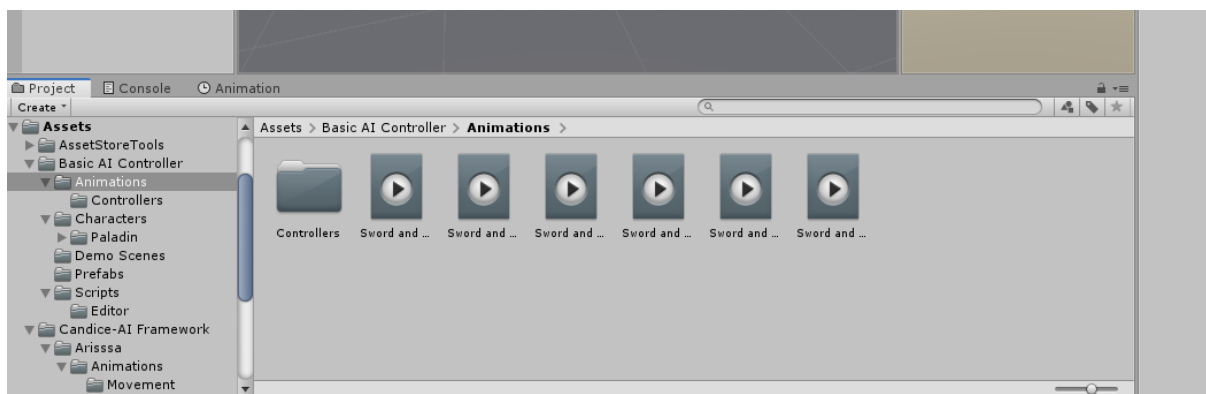


Basic AI Controller

The Basic AI Controller works best with animations. However, if you do not have animations, uncheck **Can Attack** in the inspector in order to prevent a bug when it tries to attack.

***Note:** you will require some basic knowledge on how to set up animator controllers with animations.

To use the script with animations, ensure you have animations for Idle, Walk, Run, Dame and Attack.

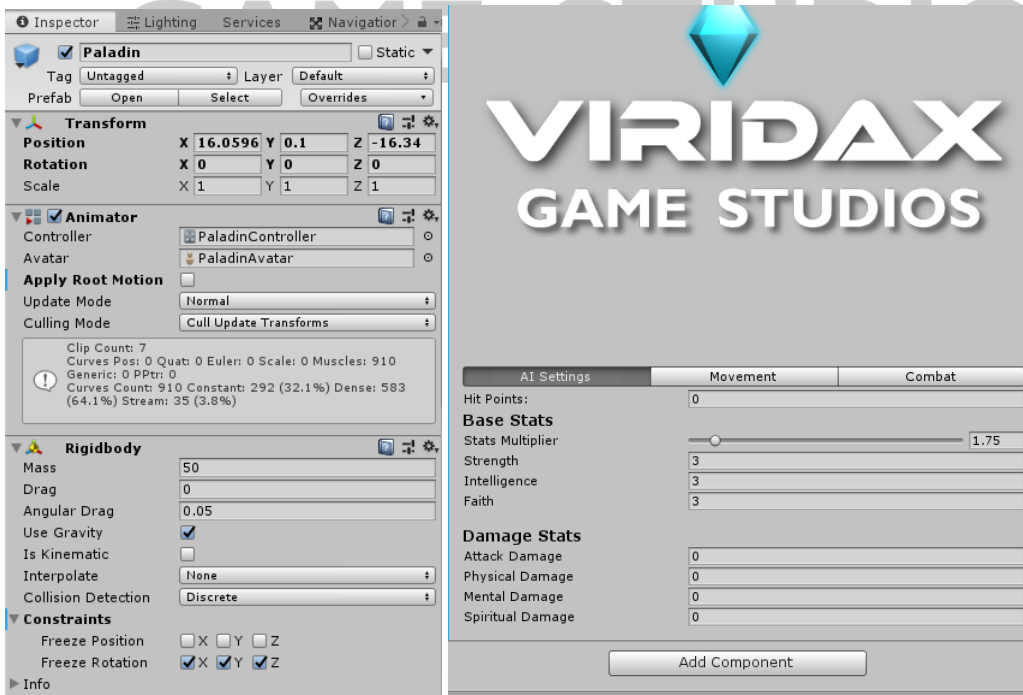


Make sure walking and running animations have "Loop Time" and "Loop Pose" enabled in order for the animations to loop seamlessly. Also, depending on what animations you use, you might have to check some other settings and also ensure that the 3D model is using the correct rig (usually it is Humanoid).



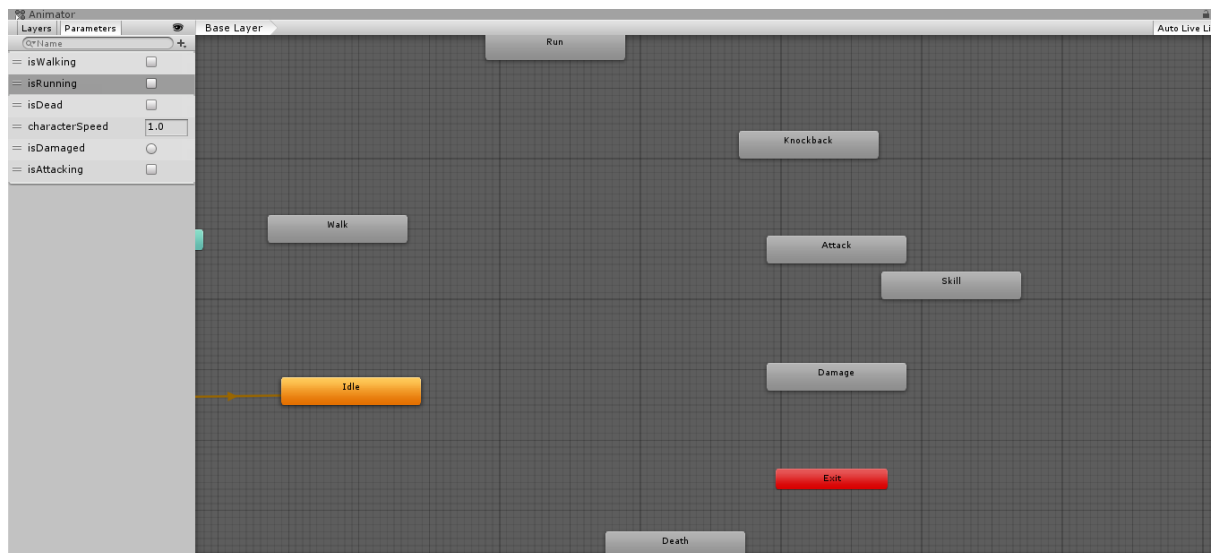
When you add the script from the Menu, it will automatically add a Rigidbody, Capsule Collider and Animator. After that, follow these steps:

1. In the Animator component, uncheck "Apply Root Motion."
2. In the Rigidbody, ensure that "Use Gravity" is checked.
3. Also in the Rigidbody go to the constraints section and check Freeze Rotation for X, Y and Z.
4. In the Capsule Collider, adjust the height, radius and centre according to your character.
5. These settings are just a guideline and you can play around with them to suit your situation.

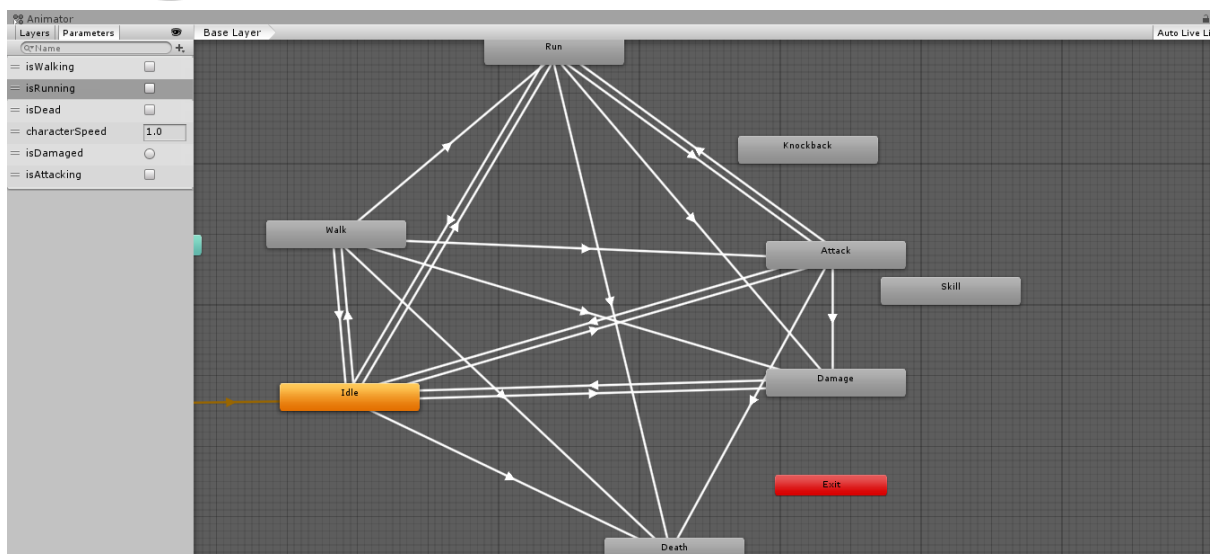


Once all of these have been set up, we can now set up the animations.

1. Create a new Animator Controller and add it to the Controller section in the Animator. In the case of the demo scene it is **Paladin Controller**.
2. Double click on your Animator Controller and then drag all the required animations into it.
3. By now you should have something looking similar to this:



4. You can now add transitions between the different animations. You will need the following conditions:
 - a. `isWalking` (Boolean), `isRunning` (Boolean), `isDead` (Boolean), `isDamaged` (Trigger), `isAttacking` (Boolean).
 - b. The demo scene (**Basic AI With and Without Animations.unity**) will show you where each transition needs to be and with what condition. Also take note that some transitions have "Has Exit Time" unchecked.
5. Your animator controller should look something like this by now:



Features

Character Class

The Basic AI Controller inherits from a class called “Character”. This class stores information/attributes about the character such as Strength, Intelligence, Faith, Attack Damage and more. It also keeps track of the Player Level.

Character Stats classes

There are several statistics but all of them are controlled by 3 primary attributes:

- Strength.
- Intelligence.
- Faith.

Each of these attributes directly affect three forms of damage.

- Physical Damage (Strength)
- Magical Damage (Intelligence)
- Spiritual Damage (Faith)

Levelling Up

The Character Stats class supports levelling up by calling the LevelUp() method. This will automatically adjust the character attributes. This feature enables the AI characters to level up and become stronger as the game progresses.

States

The controller supports different states that the character can be in. The states are

- Idle
- Follow
- Attack
- Dame
- Patrol
- Dead

By default, the controller should change between states automatically.

***Note: The character can only be in one state at a time.**

Patrol State

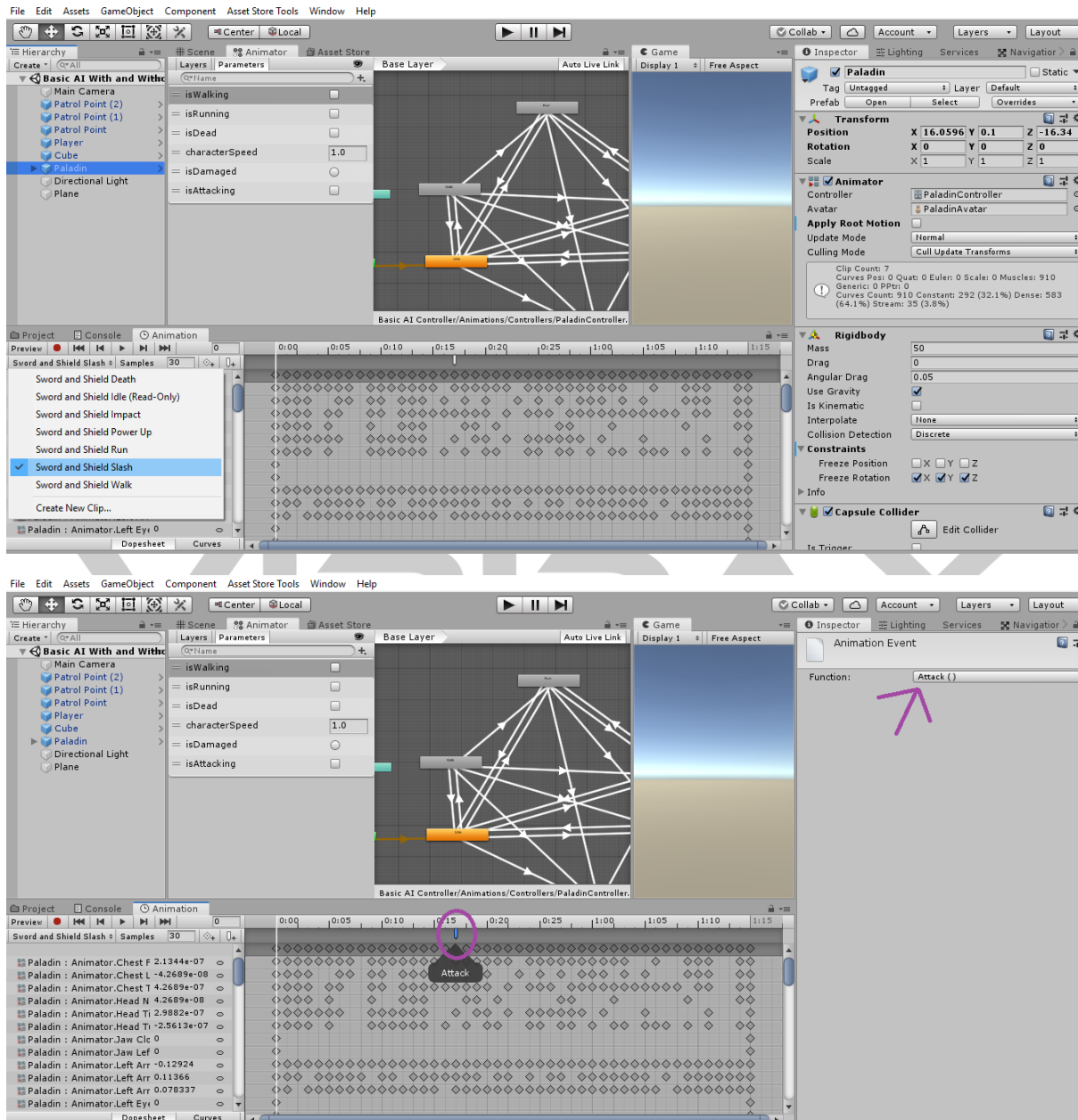
For the character to patrol between several points, add the required Patrol Points to the controller in the inspector:

The patrol points can be anything from a simple empty GameObject to more complex things. Make sure each Patrol Point is tagged as ‘PatrolPoint’.

Attack State

The AI unit can also attack and deal damage to objects around. You will need to use an animation Event to call the Attack() method in the script. To do so, follow these steps:

1. Select the 3D Model, in this case it is **Paladin**.
2. Select the **Animation Window**.
3. Select the Attack animation.
4. Add an animation Event at the point where you feel the damage should be dealt. Usually where the sword comes down and makes contact with the player.
5. Choose Attack() as the function to call.



Damage System

The Damage System works by sending a Raycast as a sphere around the character. If it collides with an object marked with any of the Enemy Tags, a message will be sent to the enemy object in order to receive damage. Ensure that the receiving object has a method with the name `ReceiveDamage()`. It should have the following method signature

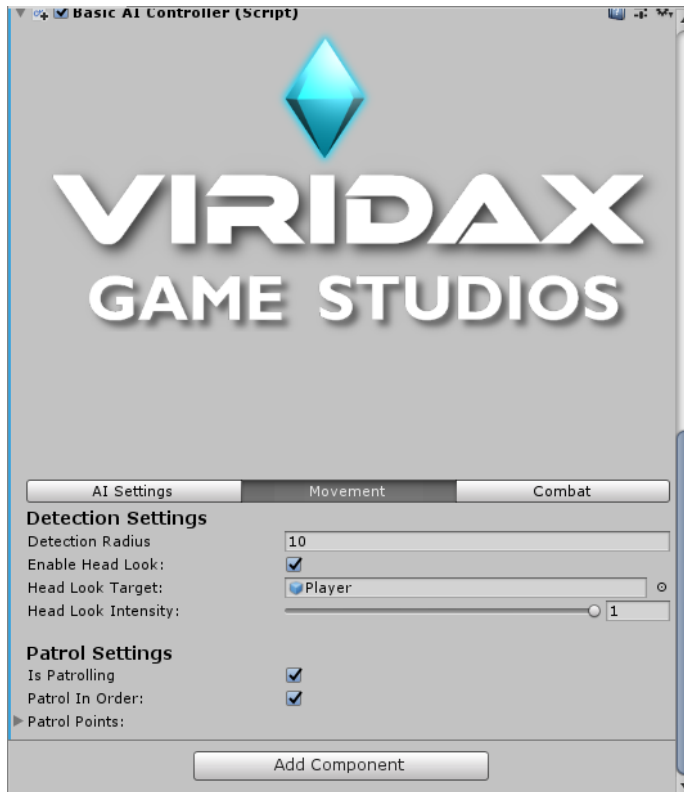
```
Public void ReceiveDamage(float damage) { }
```

Head Look

The Basic AI Controller also has a Head Look feature. This allows the character to dynamically look at any object in the game. To use this feature follow these steps:

1. Click on the Movement tab.
2. Check the “Enable Head Look” option.
3. Select a Target, you can drag and drop any gameobject into the slot.

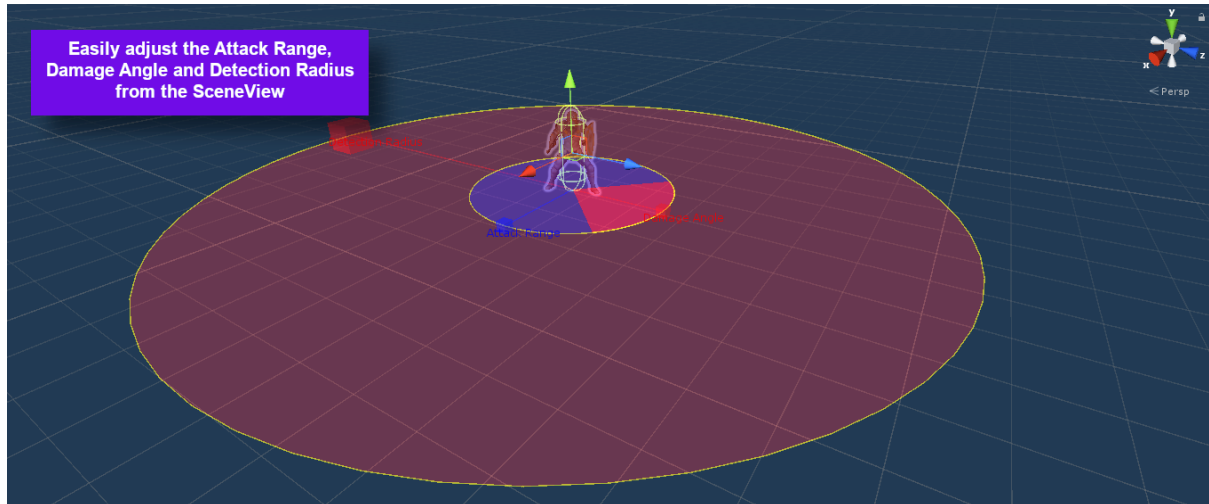
The intensity determines how intensely the character will focus on the object.



Editor and SceneView

Basic A.I Controller also has powerful editor and SceneView controls. You can adjust the properties including attack range and detection radius right from the SceneView. You can also adjust all other properties, like enemy tags, move speed and more from the inspector. There's no need to touch the

script if you don't want to.



Support

We at Viridax Game Studios strive to provide the best AAA quality video game script assets to the world. If you have any issues with this asset, suggestions, or just want to send some nice feedback, then feel free to contact us on one of our channels.

Support email: natubeast@gmail.com

Discord Server: <https://discord.gg/Tar7nxX>

Facebook: <https://www.facebook.com/viridaxgamestudios/>

References

Item	Author	Website
Paladin 3D Model	Mixamo	https://www.mixamo.com
Sword and Shield Animations	Mixamo	https://www.mixamo.com