Basic AI Controller

User Manual

Contents

VIRIDAX GAME STUDIOS

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.

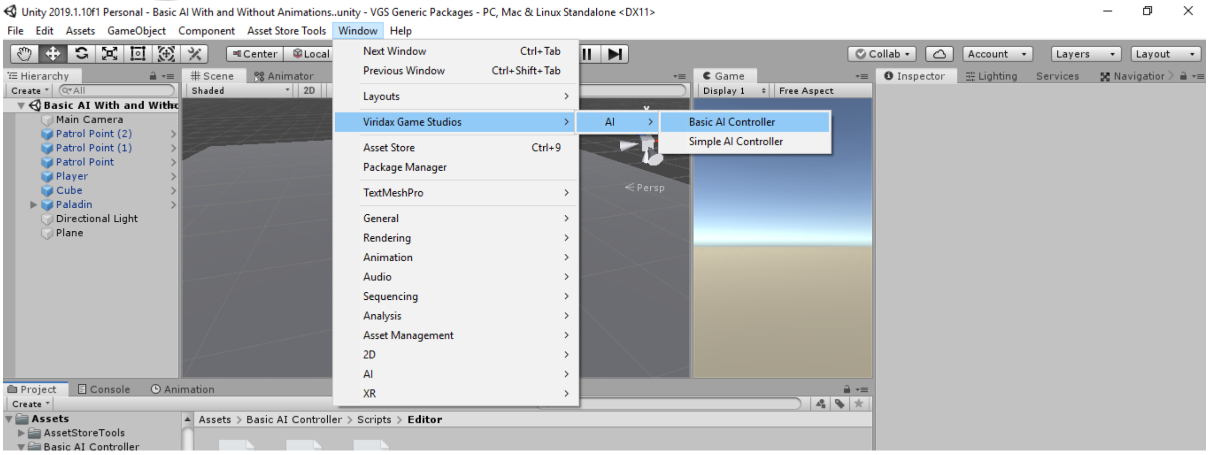
Before you begin using any of the Basic A.I controller script, make sure you have done the following first:

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())

VIRIDAX GAME STUDIOS

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.

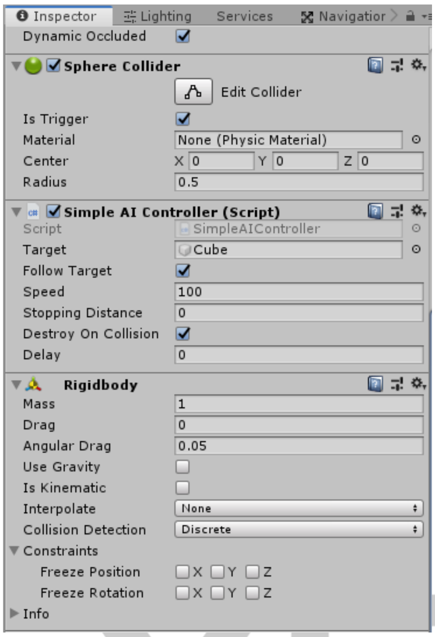
If you added the Basic AI Controller script from the menu, then it will automatically add the “Player” tag to the list of enemy tags. Note: the default reaction is to attack. If you added the script manually, you will have to do this manually from the inspector or in code.

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.



Dynamic Occluded ☑

Sphere Collider ☑

Is Trigger ☑
Material None (Physic Material)
Center X 0 Y 0 Z 0
Radius 0.5

Simple AI Controller (Script) ☑

Script SimpleAIController
Target Cube
Follow Target ☑
Speed 100
Stopping Distance 0
Destroy On Collision ☑
Delay 0

Rigidbody

Mass 1
Drag 0
Angular Drag 0.05
Use Gravity □
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position □ X □ Y □ Z
Freeze Rotation □ X □ Y □ Z

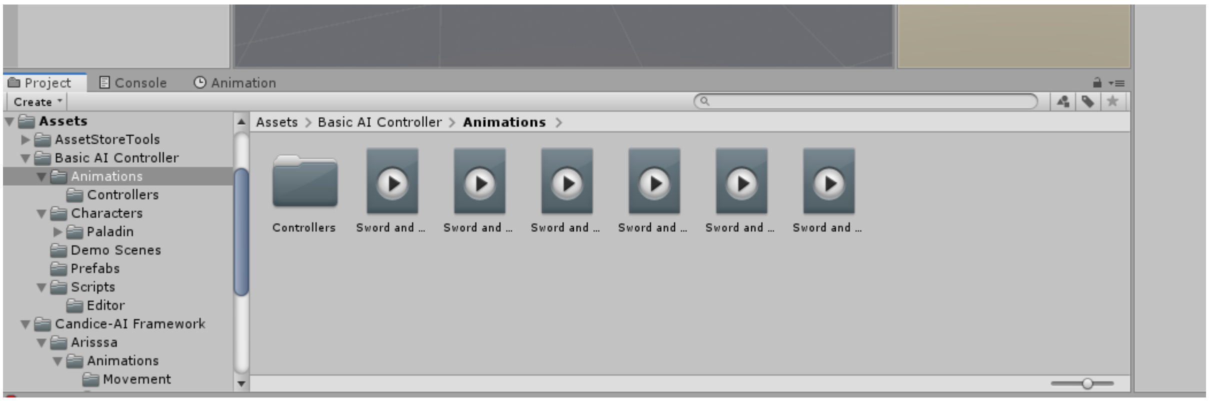
VIRIDAX GAME STUDIOS

Basic AI Controller

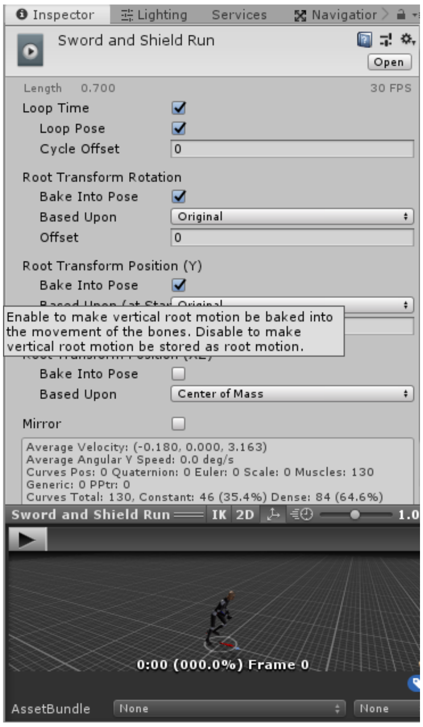
The Basic AI Controller works best with animations. However, if you do not have animations, uncheck **Can Attack** in the inspector in order to prevent a bug when it tries to attack.

***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).



Sword and Shield Run

Length 0.700 30 FPS

Loop Time ☑

Loop Pose ☑

Cycle Offset 0

Root Transform Rotation

Bake Into Pose ☑

Based Upon: Original

Offset 0

Root Transform Position (Y)

Bake Into Pose ☑

Based Upon (at Star Original

Enable to make vertical root motion be baked into the movement of the bones. Disable to make vertical root motion be stored as root motion.

Bake Into Pose □

Based Upon: Center of Mass

Mirror □

```
Average Velocity: (-0.180, 0.000, 3.163)
Average Angular Y Speed: 0.0 deg/s
Curves Pos: 0 Quaternion: 0 Euler: 0 Scale: 0 Muscles: 130
Generic: 0 PPtr: 0
Curves Total: 130, Constant: 46 (35.4%) Dense: 84 (64.6%)
```

Sword and Shield Run

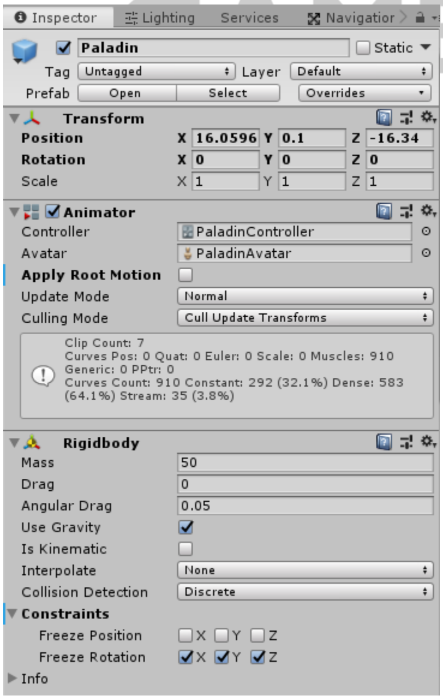
0:00 (000.0%) Frame 0

AssetBundle None None

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.

VIRIDAX GAME STUDIOS



☑ Paladin □ Static

Tag: Untagged Layer: Default

Prefab

Transform

Position X 16.0596 Y 0.1 Z -16.34

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Animator

Controller PaladinController

Avatar PaladinAvatar

Apply Root Motion □

Update Mode: Normal

Culling Mode: Cull Update Transforms

```
Clip Count: 7
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 910
Generic: 0 PPtr: 0
Curves Count: 910 Constant: 292 (32.1%) Dense: 583
(64.1%) Stream: 35 (3.8%)
```

Rigidbody

Mass 50

Drag 0

Angular Drag 0.05

Use Gravity ☑

Is Kinematic □

Interpolate: None

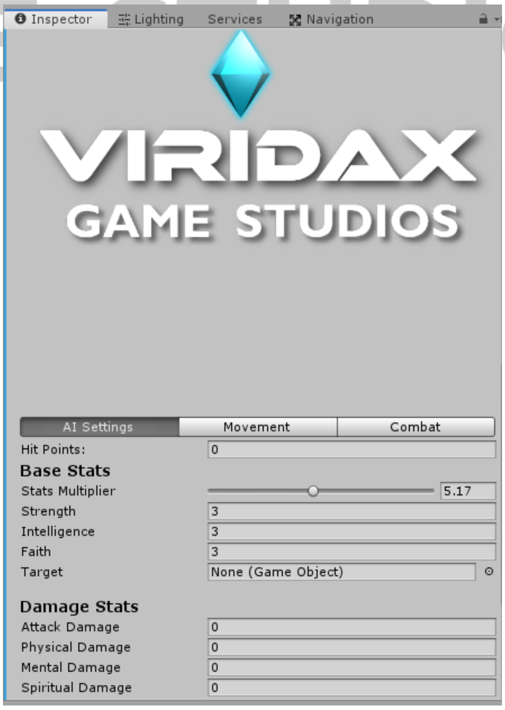
Collision Detection: Discrete

Constraints

Freeze Position □ X □ Y □ Z

Freeze Rotation ☑ X ☑ Y ☑ Z

Info



Hit Points: 0

Base Stats

Stats Multiplier 5.17

Strength 3

Intelligence 3

Faith 3

Target None (Game Object)

Damage Stats

Attack Damage 0

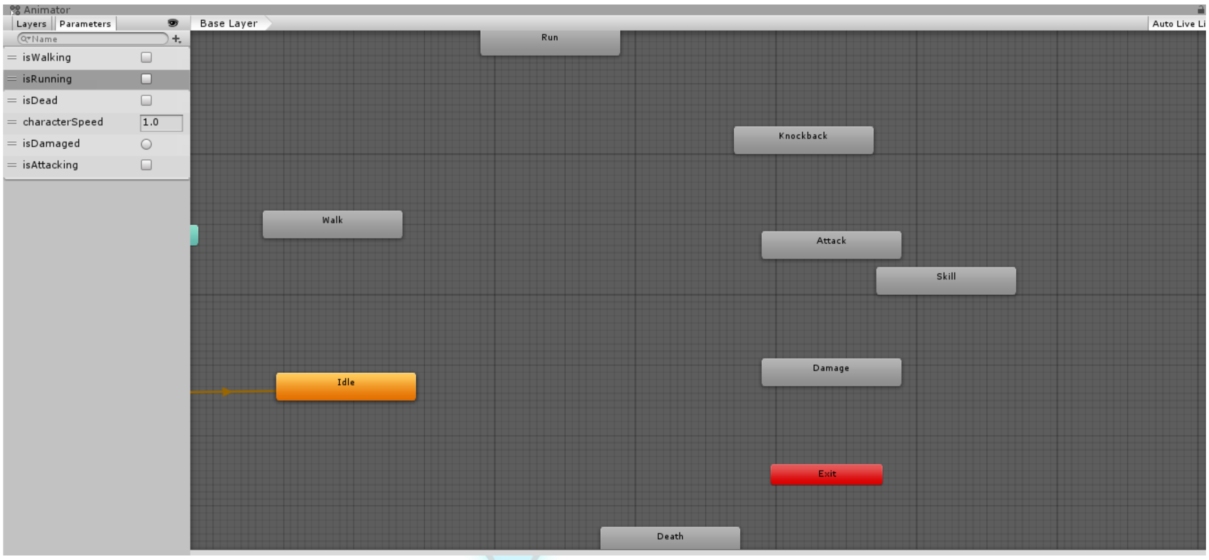
Physical Damage 0

Mental Damage 0

Spiritual Damage 0

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

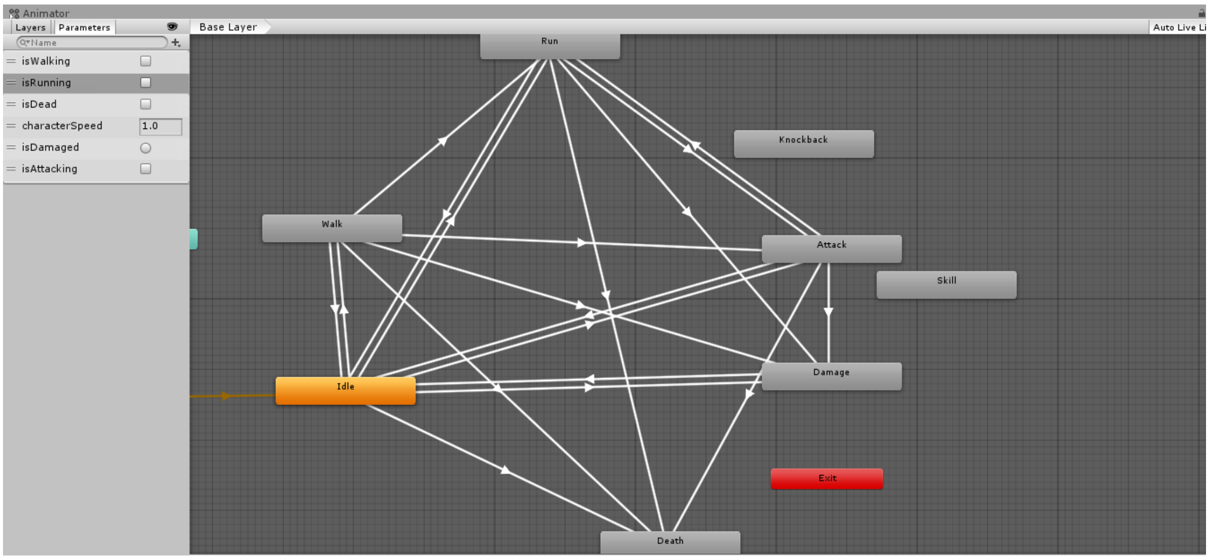
1. Create a new Animator Controller and add it to the Controller section in the Animator. In the case of the demo scene it is **Paladin Controller**.
2. Double click on your Animator Controller and then drag all the required animations into it.
3. By now you should have something looking similar to this:



isWalking □
isRunning □
isDead □
characterSpeed 1.0
isDamaged ○
isAttacking □

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:

VIRIDAX GAME STUDIOS



isWalking □
isRunning □
isDead □
characterSpeed 1.0
isDamaged ○
isAttacking □

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, using Finite State Machines. The states are

- Idle
- Follow
- Attack
- Patrol

By default, the controller should change between states automatically.

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

In order for a state to work, you must create an Action class that inherits from (FSMAction) and then assign it to the state you want. For example, create a MoveAction Action that can be assigned to the Follow State and Patrol State (since they both require the agent to move).

Patrol State

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

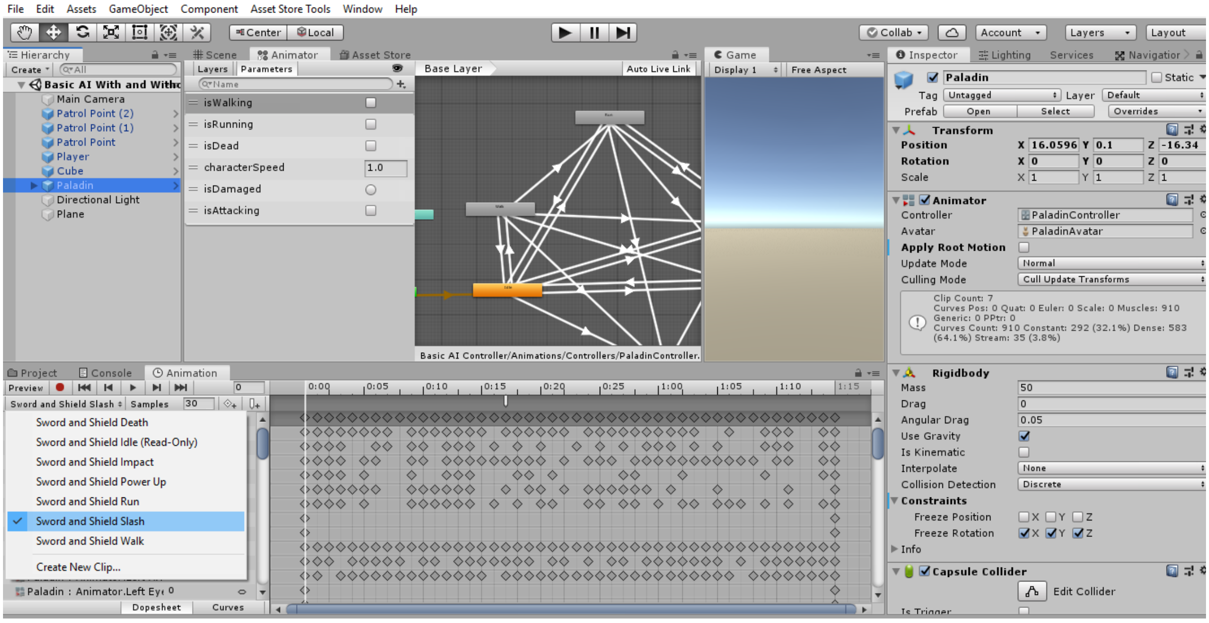
The patrol points can be anything from a simple empty GameObject to more complex things. Make sure each Patrol Point is tagged as ‘PatrolPoint’.

VIRIDAX
GAME STUDIOS

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.



isWalking □
isRunning □
isDead □
characterSpeed 1.0
isDamaged ○
isAttacking □

Basic AI Controller/Animations/Controllers/PaladinController.

Sword and Shield Slash · Samples 30

- Sword and Shield Death
- Sword and Shield Idle (Read-Only)
- Sword and Shield Impact
- Sword and Shield Power Up
- Sword and Shield Run
- ✓ Sword and Shield Slash
- Sword and Shield Walk
- Create New Clip...

Paladin : Animator.Left Eye 0

☑ Paladin □ Static
Tag Untagged Layer Default
Prefab Open Select Overrides

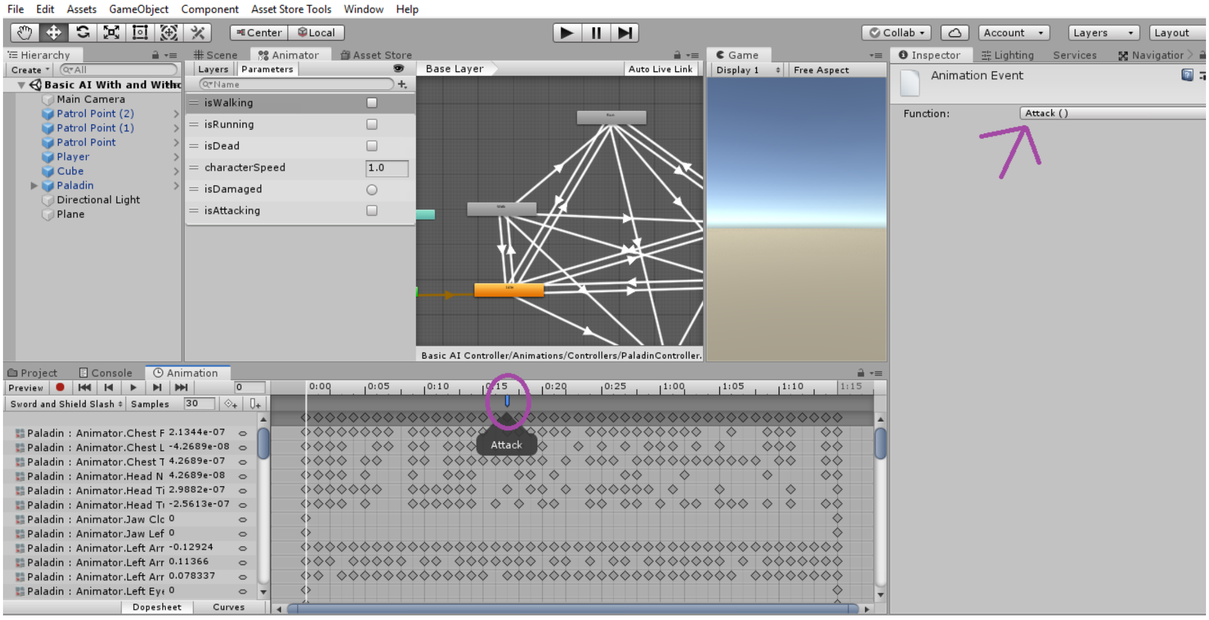
Transform	X	Y	Z
Position	16.0596	0.1	-16.34
Rotation	0	0	0
Scale	1	1	1

☑ Animator
Controller PaladinController
Avatar PaladinAvatar
Apply Root Motion □
Update Mode Normal
Culling Mode Cull Update Transforms

Clip Count: 7
Curves Pos: 0 Quat: 0 Euler: 0 Scale: 0 Muscles: 910
Generic: 0 PPtr: 0
Curves Count: 910 Constant: 292 (32.1%) Dense: 583 (64.1%) Stream: 35 (3.8%)

Rigidbody
Mass 50
Drag 0
Angular Drag 0.05
Use Gravity ☑
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position □X □Y □Z
Freeze Rotation ☑X ☑Y ☑Z
Info

☑ Capsule Collider
Edit Collider
Is Trigger □



isWalking □
isRunning □
isDead □
characterSpeed 1.0
isDamaged ○
isAttacking □

Basic AI Controller/Animations/Controllers/PaladinController.

Sword and Shield Slash · Samples 30

Paladin : Animator.Chest F 2.1344e-07
Paladin : Animator.Chest L -4.2689e-08
Paladin : Animator.Chest T 4.2689e-07
Paladin : Animator.Head N 4.2689e-08
Paladin : Animator.Head Ti 2.9882e-07
Paladin : Animator.Head Ti -2.5613e-07
Paladin : Animator.Jaw Clc 0
Paladin : Animator.Jaw Lef 0
Paladin : Animator.Left Arr -0.12924
Paladin : Animator.Left Arr 0.11366
Paladin : Animator.Left Arr 0.078337
Paladin : Animator.Left Ey 0

Animation Event
Function: Attack ()

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

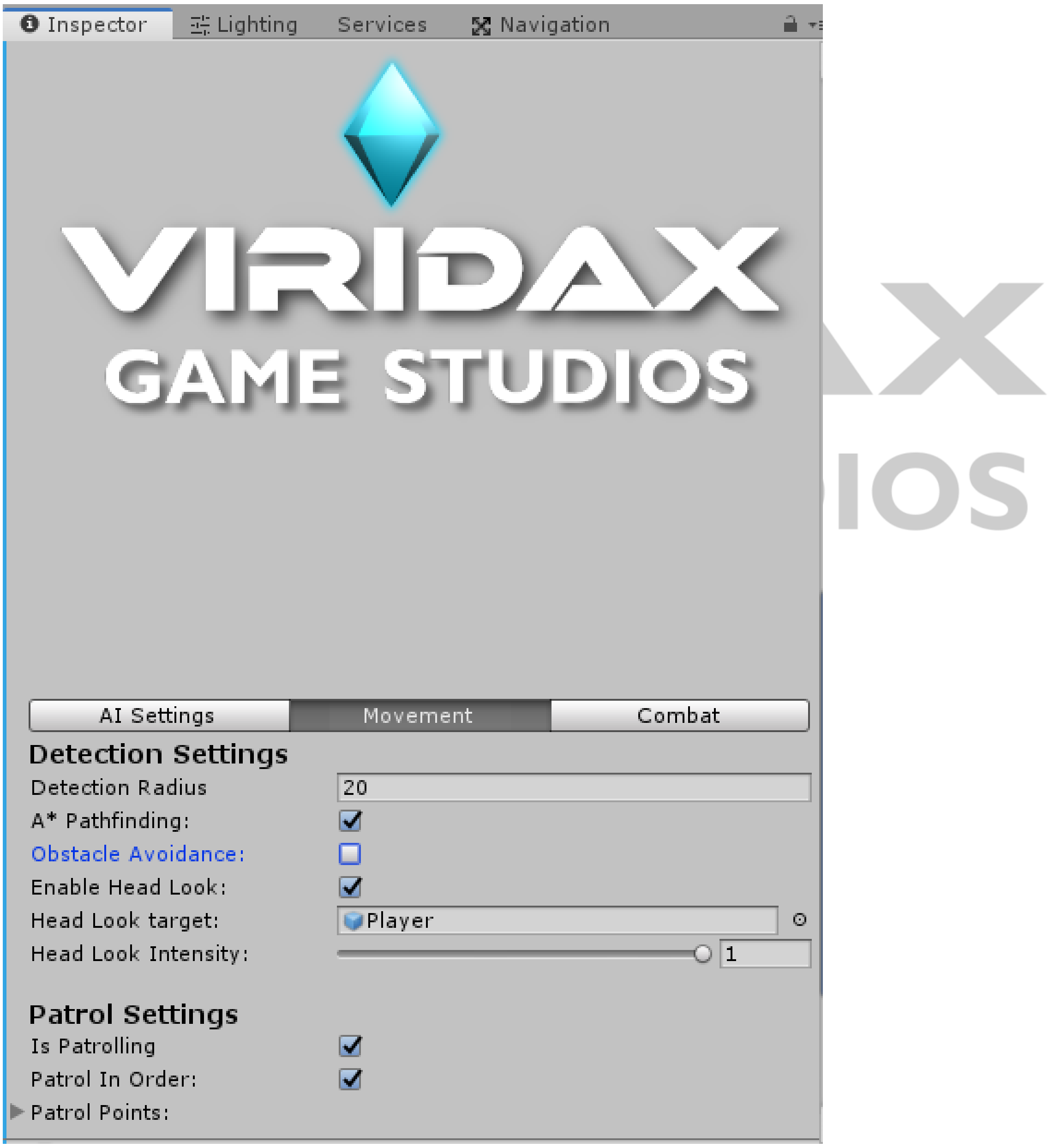
Public void ReceiveDamage(float damage) { }

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.



Detection Settings

Detection Radius 20

A* Pathfinding: ☑

Obstacle Avoidance: □

Enable Head Look: ☑

Head Look target: Player

Head Look Intensity: 1

Patrol Settings

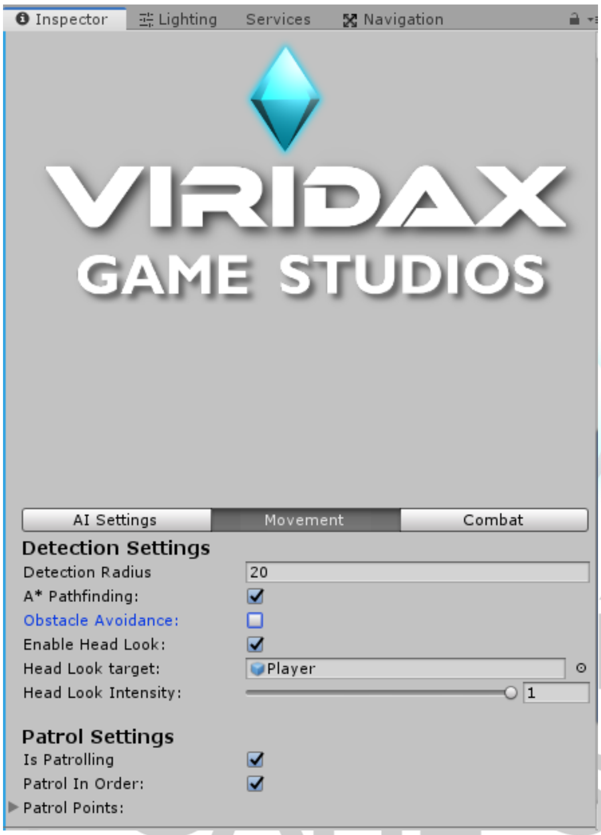
Is Patrolling ☑

Patrol In Order: ☑

Patrol Points:

Movement

The AI controller supports movement with two primary ways of evading obstacles. The first and simplest is using basic Obstacle Avoidance and the second is using Grid-Based A* Pathfinding. In order to enable any of these features, navigate to the Movement Tab in the Inspector controls for the AI Controller and check the relative options. **Note: If both A* and Obstacle Avoidance are both activated, A* will take preference due to its increased accuracy.**



Detection Settings

Detection Radius 20

A* Pathfinding: ☑

Obstacle Avoidance: □

Enable Head Look: ☑

Head Look target: Player

Head Look Intensity: 1

Patrol Settings

Is Patrolling ☑

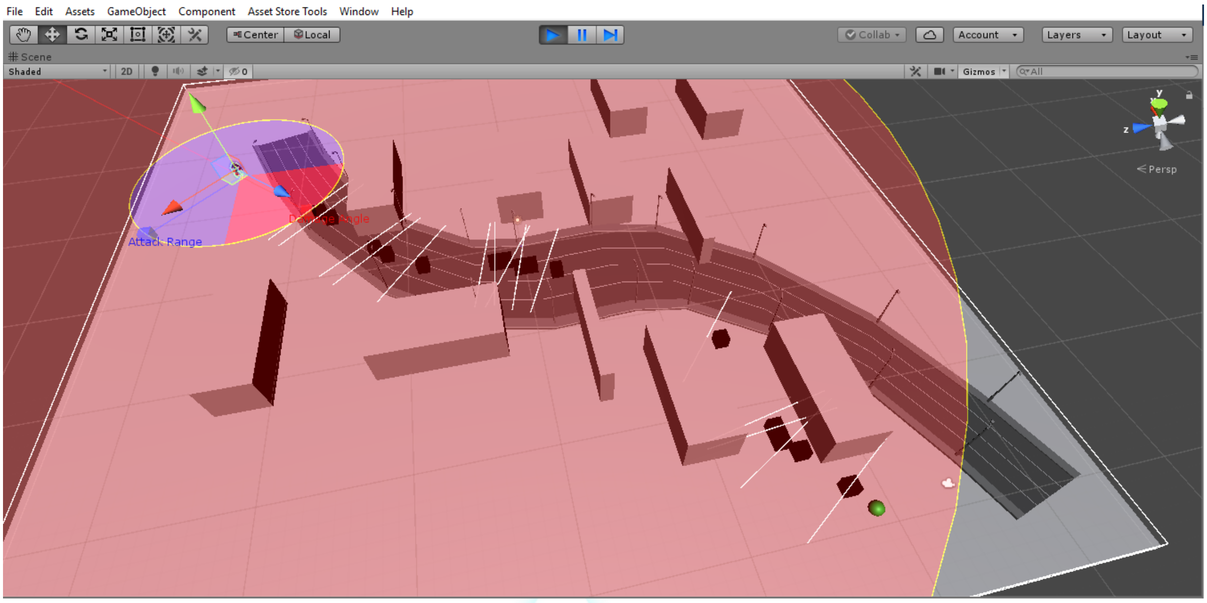
Patrol In Order: ☑

▸ Patrol Points:

Obstacle Avoidance

This allows the AI to evade immediate obstacles, within a certain range. This algorithm is most efficient in straight line paths with a few random obstacles. However, it can also be used in more complex scenes, but efficiency is not guaranteed.

A* Pathfinding



This algorithm allows the agent to find the shortest path to the target.

To make it work in your own scenarios, follow these steps:

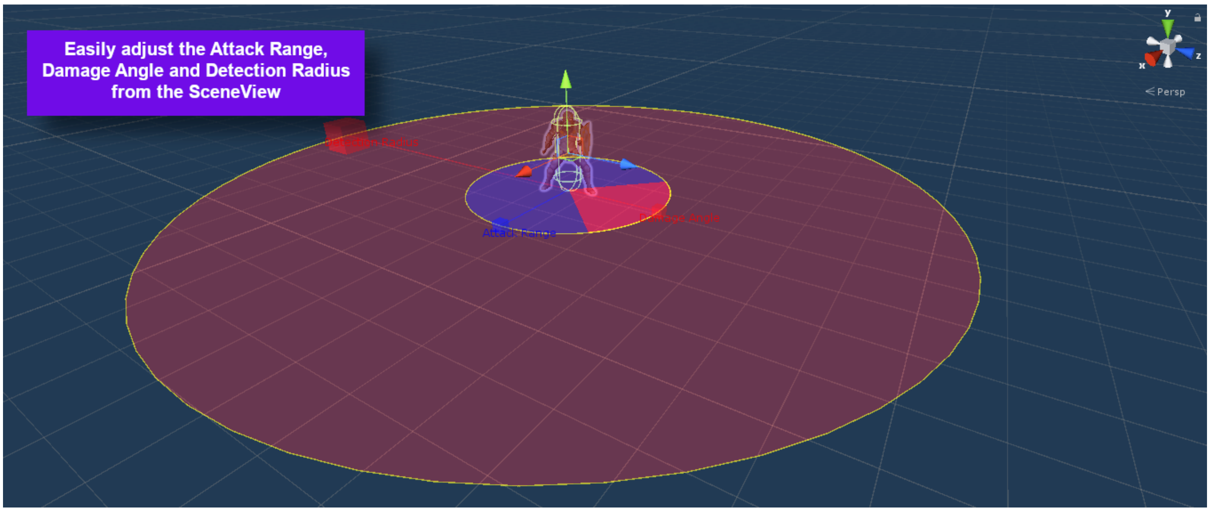
1. Ensure you have an A* Brain GameObject
    a. Create an empty game object and rename it to A* Brain.
2. Add a Grid Component to the Game Object.
3. Tweak the settings to fit your terrain and define an unwalkable mask. Unwalkable mask is the Layer which the algorithm will define as unwalkable, and thus the AI will never walk on terrain/gameobjects with that Layer.
4. Define Walkable Regions.
    a. For each walkable region, you must specify a Terrain Mask (basically a Layer that will be assigned to a terrain/gameobject) and a movement penalty for that terrain. This allows the AI to “prefer” certain terrain types over others.
5. Add a Pathfinding and PathRequestManager components to the Game Object.

VIRIDAX GAME STUDIOS

Attach the Unit script/Component to the GameObject that has the AI controller. Now you can call StartFindingPath() and StopFindingPath() respectively and it will automatically calculate the best path and move the agent.

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.


Easily adjust the Attack Range, Damage Angle and Detection Radius from the SceneView
Detection Radius
Attack Range
Damage Angle
y
x
z
Persp

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.

VIRIDAX GAME STUDIOS

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