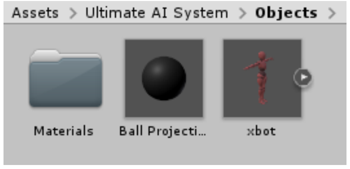
This is a quick guide which will help you configure the basic settings of the "Ultimate AI System". Multiple video tutorials can be found online.

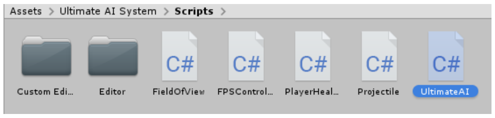
Contents

Getting started

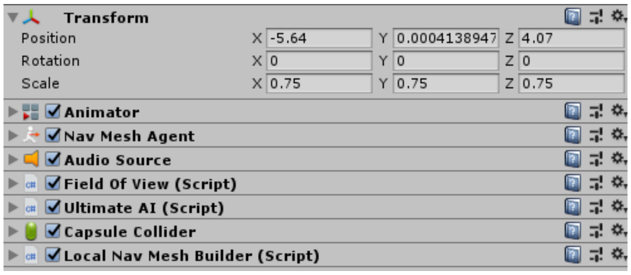
Import a 3D model of your choice to use as an AI object. It could be anything – from a cube to an ultra-realistic character. Humanoid models are recommended as they have bones and the AI will look much more realistic when playing the animations. For a quick startup there is a model included, called “xbot” which can be found in the “Objects” subfolder.



Once you have chosen your model simply drag it to the scene window and position it anywhere you like. Once you’ve done that you can go to the “Scripts” folder and append the “Ultimate AI System” script to your AI model.



Before you start tweaking the script’s settings, you’ll notice that more components are added. This is ok – you’ll need them later! An empty game object called “Pathfinding” will be added as well.



Transform

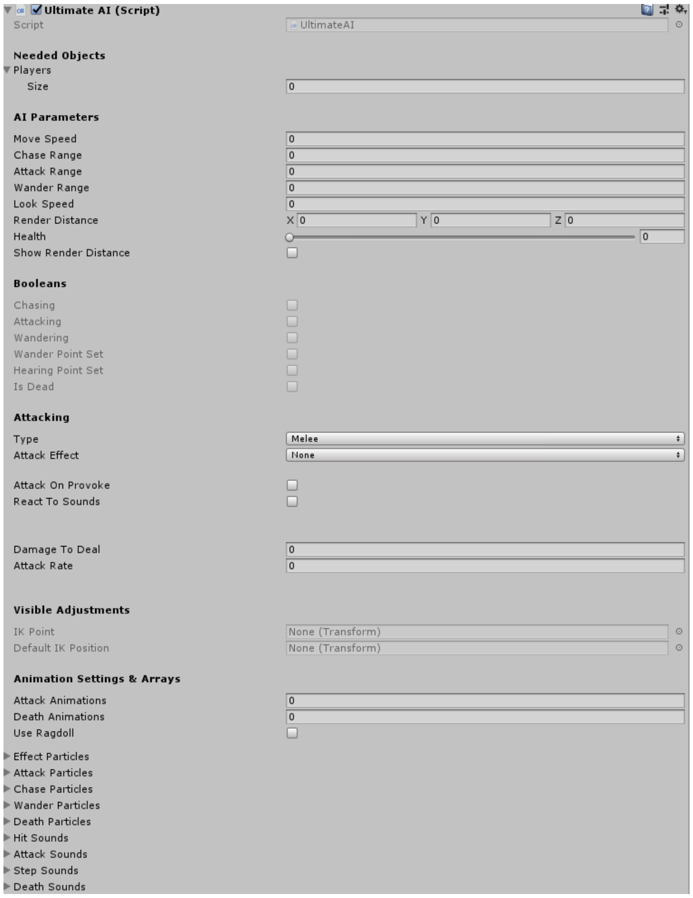
Position	X -5.64	Y 0.0004138947	Z 4.07
Rotation	X 0	Y 0	Z 0
Scale	X 0.75	Y 0.75	Z 0.75

☑ Animator
☑ Nav Mesh Agent
☑ Audio Source
☑ Field Of View (Script)
☑ Ultimate AI (Script)
☑ Capsule Collider
☑ Local Nav Mesh Builder (Script)

Then for each player you have in your game you will need to add the “PlayerHealth” component which can also be found in the “Scrips” folder. Make sure to set a “Health” value and leave the rest for later.

How to Set up a Melee AI

1. Drag and drop all of your players objects in the “Players” list under “Needed Objects”.



Ultimate AI (Script)	
Script	UltimateAI
Needed Objects	
Players	
Size	0
AI Parameters	
Move Speed	0
Chase Range	0
Attack Range	0
Wander Range	0
Look Speed	0
Render Distance	X 0 Y 0 Z 0
Health	0
Show Render Distance	□
Booleans	
Chasing	□
Attacking	□
Wandering	□
Wander Point Set	□
Hearing Point Set	□
Is Dead	□
Attacking	
Type	Melee
Attack Effect	None
Attack On Provoke	□
React To Sounds	□
Damage To Deal	0
Attack Rate	0
Visible Adjustments	
IK Point	None (Transform)
Default IK Position	None (Transform)
Animation Settings & Arrays	
Attack Animations	0
Death Animations	0
Use Ragdoll	□
Effect Particles	
Attack Particles	
Chase Particles	
Wander Particles	
Death Particles	
Hit Sounds	
Attack Sounds	
Step Sounds	
Death Sounds	

2. Next assign all the variables inside the “AI Parameters” section as you’d like them to be. You will first have to define a speed for your AI to use when moving. Then assign a chase range – this is from how far the AI will be able to chase a player if it is in its line of sight. Same thing applies for both the “Attack Range” and the “Wander Range” (“Wander Range” is how far the AI can get a random point to go to when there are not any players it can attack or chase). Each of those ranges will be visualized as a circle surrounding the AI and will have a different color from the rest.

NOTE: The relationship between the “Chase Range” and the “Attack Range” must always be like this:

Chase Range > Attack Range

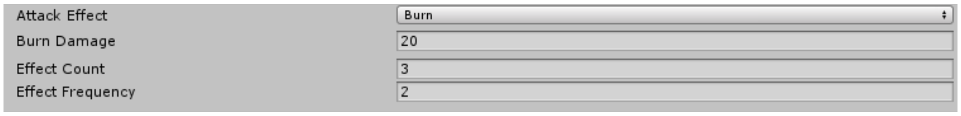
Then you will have to set a rotation speed or as it is called in the script “Look Speed”. We recommend you to assign a higher value. Something like 180. Next choose how far the AI is able to calculate its pathing by setting up the “Render Distance”. You get to choose a value for each axis (X, Y, Z). **NOTE:** There must not be any range with higher value compared to any of the axis in the “Render Distance” variable divided by 2. Now insert a value for the health of the AI. The next setting has no effect on the AI’s behaviour. If it is toggled it will just visualize the “Render Distance” inside the editor window.

3. The “Booleans” section is just to visualize what the current state of the AI is. The next section is “Attacking” and the very first variable from it is of crucial importance. Depending on what type you choose your AI to be (under “Type”) it will behave in a different way. Melee will make it perform close-ranged (melee) attacks, Ranged will make it shoot at the player with custom projectiles and NPC will make it just wander around and when the player is close it will look towards the player. But right now, we are creating a melee-behaving one so let’s leave it as is.
4. You then get to choose whether you want the AI to affect the player when attacking. Those will be applied when an attack got executed. Here’s the list of options you can choose from:

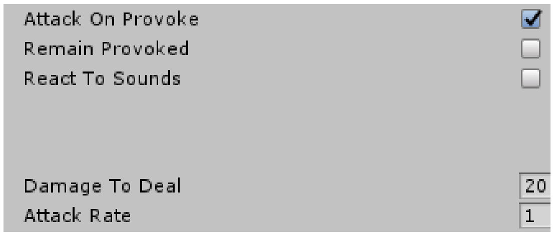
   A. Poison – the player is poisoned when hit by the AI. This means the player will take some extra damage from time to time because of the poison feature of the AI’s attack. You also get to set custom particles to make the feel as real as possible.

B. Burning – similar to the poison the fire ability sets the player on fire and deals damage from time to time.
C. Slow – this ability simply slows down the player for a given amount of time.
D. Freeze – this ability prevents the player from moving until the effect is gone.
E. Stun – stunning blurs the player's visibility making the gameplay harder until the effect is gone.
F. Healing – when the AI performs an attack not only does it deal damage to the player, but it also regenerates some health for itself.
G. None – if you want to go with a normal and less powerful AI we've got it covered!

You will then need to assign the effect value which will vary based on what effect you chose and how many times the effect will repeat in the "Effect Count" field. Then if your effect supports that setting you will see "Effect Frequency" field. If the "Effect Count" is equal to 1 this is how long it will last or else this will be the delay between executing the next effect (the AI will execute the same effect until it reaches the "Effect Count" value).

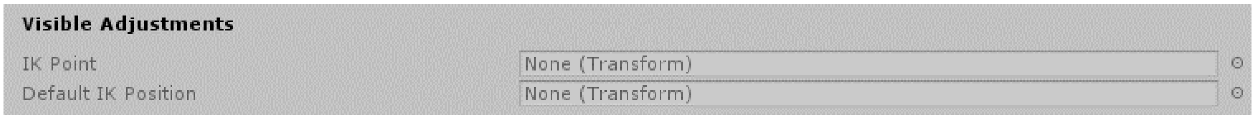


Attack Effect	Burn
Burn Damage	20
Effect Count	3
Effect Frequency	2



Attack On Provoke	☑
Remain Provoked	□
React To Sounds	□
Damage To Deal	20
Attack Rate	1

5. You then get to choose if you want your AI to be passive until provoked/hit or be always hostile. If you do not check the "Remain Provoked" setting your AI will now forget that it is "angry" at a player once it is far enough for the AI not to be able to see it or you can make the AI unforgiving by toggling this.
6. You can now choose to turn on the "React To Sounds" setting. However, do not do that yet. Let's first set the AI and then take a deeper look at the HEARING section.
7. You then need to select a damage amount to be taken from the player's health when being successfully attacked next to "Damage to Deal".
8. The "Attack Rate" is the time in seconds that will be used as a pause between the attacks. This prevents the AI from spamming attacks unlimitedly.
9. Next on the list you will see the "Visible Adjustments" tab. Those are the inverse kinematics settings (IK) and will only be available if your Unity Version is higher or equal to "2019.1". If you are on an older version those settings will be inactive (colored in grey) like so:



Visible Adjustments

IK Point	None (Transform)
Default IK Position	None (Transform)

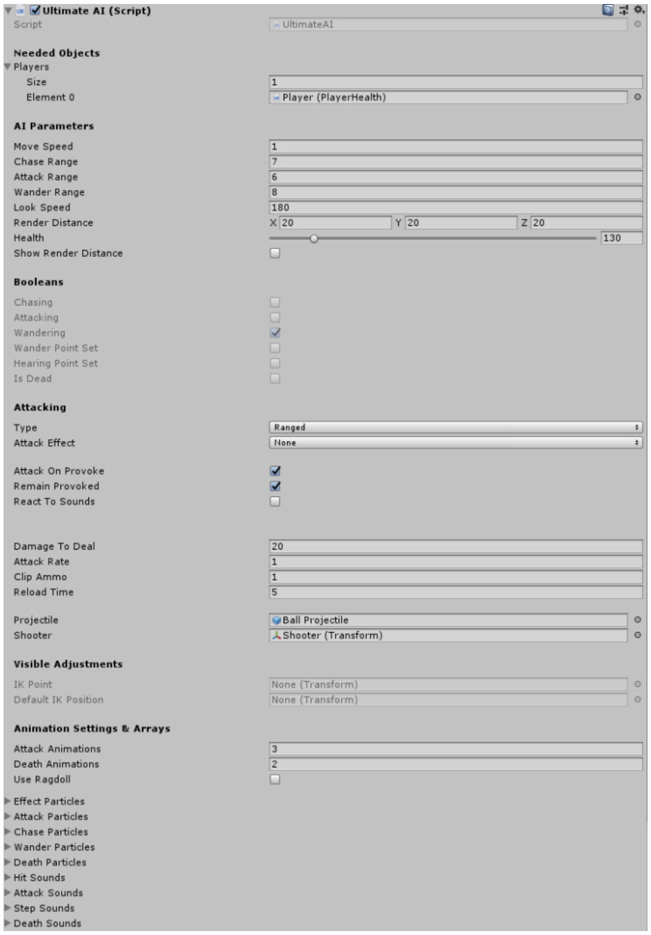
If your version supports IK and you want your AI to have IK on its head then wait a bit. Once we are done setting the AI's attacking behavior you will be free to check out the INVERSE KINEMATICS section. But before that you need to get trough the LINE OF SIGHT section.

How to Set up a Ranged AI

1. Drag and drop all of your players objects in the “Players” list under “Needed Objects”.
2. Next assign all the variables inside the “AI Parameters” section as you’d like them to be. You will first have to define a speed for your AI to use when moving. Then assign a chase range – this is from how far the AI will be able to chase a player if it is in its line of sight. Same thing applies for both the “Attack Range” and the “Wander Range” (“Wander Range” is how far the AI can get a random point to go to when there are not any players it can attack). Each of those ranges will be visualized as a circle surrounding the AI and will have a different color from the rest.

NOTE: The relation between the “Chase Range” and the “Attack Range” must always be like this: Chase Range > Attack Range

Then you will have to set a rotation speed or as it is called in the script “Look Speed”. We recommend you to assign a higher value. Something like 180. Next choose how far the AI is able to calculate its pathing by setting up the “Render Distance”. You get to choose a value for each axis (X, Y, Z). **NOTE:** There must not be any range with higher value compared to any of the axis in the “Render Distance” variable divided by 2. Now insert a value for the health of the AI. The next setting has no effect on the AI’s behavior. If it is toggled it will just visualize the “Render Distance” inside the editor window.



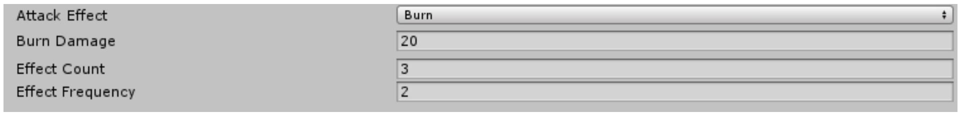
☑ Ultimate AI (Script)

Script	UltimateAI
Needed Objects	
Players	
Size	1
Element 0	Player (PlayerHealth)
AI Parameters	
Move Speed	1
Chase Range	7
Attack Range	6
Wander Range	8
Look Speed	180
Render Distance	X 20 Y 20 Z 20
Health	130
Show Render Distance	□
Booleans	
Chasing	□
Attacking	□
Wandering	☑
Wander Point Set	□
Hearing Point Set	□
Is Dead	□
Attacking	
Type	Ranged
Attack Effect	None
Attack On Provoke	☑
Remain Provoked	☑
React To Sounds	□
Damage To Deal	20
Attack Rate	1
Clip Ammo	1
Reload Time	5
Projectile	Ball Projectile
Shooter	Shooter (Transform)
Visible Adjustments	
IK Point	None (Transform)
Default IK Position	None (Transform)
Animation Settings & Arrays	
Attack Animations	3
Death Animations	2
Use Ragdoll	□

3. The “Booleans” section is just to visualize what the current state of the AI is.
4. Now let’s take a look at the “Attacking” section. Right away you should see the “Type” setting. Set this to “Ranged” because we are about to create a ranged AI unit. You then get to choose whether you want the AI to affect the player when attacking. Those will be applied when an attack got executed. Here’s the list of options you can choose from:
   A. Poison – the player is poisoned when hit by the AI. This means the player will take some extra damage from time to time because of the poison feature of the AI. You also get to set custom particles to make the feel as real as possible.
   B. Burning – similar to the poison the fire ability sets the player on fire and deals damage from time to time.
   C. Slow – this ability simply slows down the player for a given amount of time.
   D. Freeze – this ability prevents the player from moving until the effect is gone.
   E. Stun – stunning blurs the player’s visibility making the gameplay harder until the effect is gone.

F. Healing – when the AI performs an attack not only does it deal damage to the player, but it also regenerates some health for itself.
G. None – if you want to go with a normal and less powerful AI, we've got it covered!

You will then need to assign the effect value which will vary based on what effect you chose and how many times the effect will repeat in the "Effect Count" field. Then if your effect supports that setting you will see "Effect Frequency" field. If the "Effect Count" is equal to 1 this is how long it will last or else this will be the delay between executing the next effect (the AI will execute the same effect until it reaches the "Effect Count" value).



Attack Effect	Burn
Burn Damage	20
Effect Count	3
Effect Frequency	2

5. You then get to choose if you want your AI to be passive until provoked/hit or be always hostile. If you do not check the "Remain Provoked" setting your AI will now forget that it is "angry" at a player once it is far enough for the AI not to be able to see it or you can make the AI unforgiving by toggling this. For now leave the "React To Sounds" setting once you are done here, we will cover it later in the HEARING section. Now chose how much damage to deal and the delay between each attack in seconds right next to the "Attack Rate" variable. Now assign a "Clip Ammo" – how much ammo each clip of the AI has and "Reload Time" – once again in seconds. Leave the "Projectile" field empty for now and check out the PROJECTILES section once we set the next 2 settings. For a "Shooter" create an empty object and position it on the tip of the barrel of your AI's weapon of choice. This is where all the projectiles will be coming from. Now create one more empty game object and put it in the center of your player. Once you do that remember to reference it in the "Player Center" field which can be found in the "Player Health" script on your player and NOT in the "Ultimate AI" one. And last but not least have a look at how you can set up your own projectile in the PROJECTILES section. Then you can play around with the RAGDOLLS, INVERSE KINEMATICS and the VISUAL IMPROVEMENTS.

How to Set up an NPC

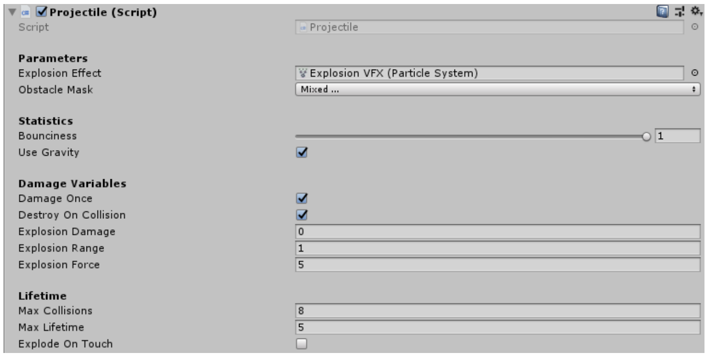
Before doing anything else go to the setting "Type" – and change it to NPC. This will remove most of the settings making your life much easier. So, let's have a look at what's left.

1. Drag and drop all of your players objects in the "Players" list under "Needed Objects" tab.
2. You will now have to define a speed for your AI to use when moving. Then assign a chase range – this is from how far the AI will be able to see a player if it is in its line of sight and start looking at it. Same thing applies for the "Wander Range" ("Wander Range" is how far the AI can get a random point to go to when there are not any players it can approach).
3. Then you will have to set a rotation speed or as it is called in the script "Look Speed". We recommend you to assign a higher value. Something like 180. Next choose how far the AI is able to calculate its pathing by setting up the "Render Distance". You get to choose a value for each axis (X, Y, Z). **NOTE:** There must not be any range with higher value compared to any of the axis in the "Render Distance" variable divided by 2. Now insert a value for the health of the AI. The next setting has no effect on the AI's behaviour. If it is toggled it will just visualize the "Render Distance" inside the editor window.

And you are done. You should now have a look at the LINE OF SIGHT section to finish your AI. Then go ahead and add some sounds or particles following the instructions in the ANIMATIONS SOUNDS & VFX section below, add some ragdoll physics again by having a look at the RAGDOLL section or maybe put some inverse kinematics (IK) on it.

Setting Projectiles

Alright, so currently our AI has everything but a projectile to shoot, which means it is not ready to be set to range attacking. we've provided a readily available projectile which is basically a ball with the "Projectile" script we made. You can use any model. Simply add the "Projectile" script. So now let's have a look at the settings. First of all, you will need to attach the not-finished-yet projectile to the "Ultimate AI System" script inside the AI object. Next assign an explosion effect (particle prefab) if the projectile will explode on touch. Then select everything **BUT** the player inside the "Obstacle Mask" variable. You can then play around with the other settings and you are done! You should now have a look at the LINE OF SIGHT section to finish your AI. Then go ahead and add some sounds or particles following the instructions in the ANIMATIONS SOUNDS & VFX section below, add some ragdoll physics again by having a look at the RAGDOLL section or maybe put some inverse kinematics (IK) on it.

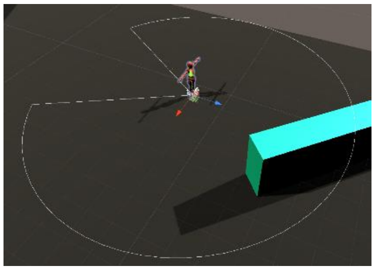


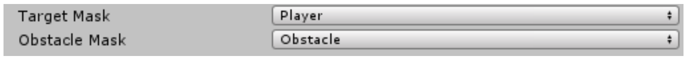
☑ Projectile (Script)

Script	Projectile
Parameters	
Explosion Effect	Explosion VFX (Particle System)
Obstacle Mask	Mixed ...
Statistics	
Bounciness	1
Use Gravity	☑
Damage Variables	
Damage Once	☑
Destroy On Collision	☑
Explosion Damage	0
Explosion Range	1
Explosion Force	5
Lifetime	
Max Collisions	8
Max Lifetime	5
Explode On Touch	□

Setting Line of sight

The first setting – "View Angle" configures the maximum field of view of the object within its view range. An angle around 300° is recommended as this will give the player the opportunity to sneak behind the AI without being seen. **NOTE:** Everytime you change the angle the graphic will be adjusted showing you the blind spots of the object. You then need to assign 2 different layers to the player and the obstacles and then select the right layers in the "Field of View" script. ***Pro tip:*** instead of selecting a single layer for "Obstacle Mask" you can simply select every layer APART from the player layer.



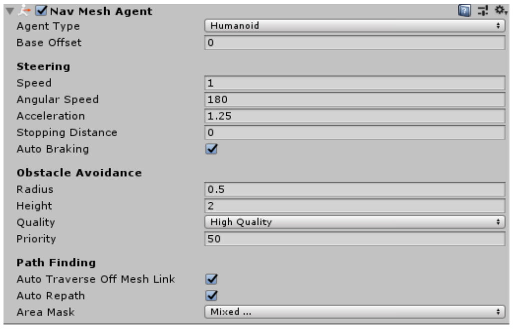


Target Mask	Player
Obstacle Mask	Obstacle

However your AI is still not done yet. Last thing you need to do is go to the PATHFINDING section.

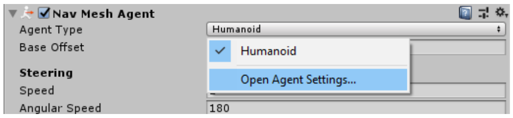
Setting Pathfinding

Once you added the "Ultimate AI" script a new component called "Nav Mesh Agent" is added as well to the AI object. If you open it up you will see 3 important settings: "Agent Type", "Radius" and "Height". The "Agent Type" is the configuration that will be selected when pathfinding. If you click on it and then select "Open Agent Settings" a menu will open.



☑ Nav Mesh Agent

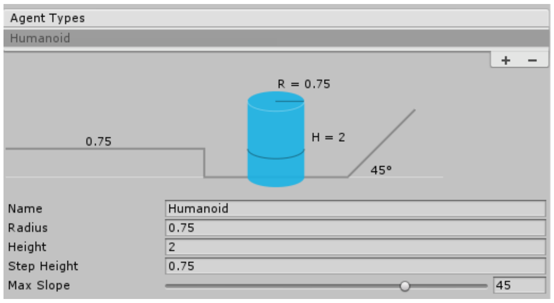
Agent Type	Humanoid
Base Offset	0
Steering	
Speed	1
Angular Speed	180
Acceleration	1.25
Stopping Distance	0
Auto Braking	☑
Obstacle Avoidance	
Radius	0.5
Height	2
Quality	High Quality
Priority	50
Path Finding	
Auto Traverse Off Mesh Link	☑
Auto Repath	☑
Area Mask	Mixed ...



☑ Nav Mesh Agent

Agent Type	Humanoid
Base Offset	
Steering	
Speed	
Angular Speed	180

In here you can add mutliple profiles for each AI. Assign accurate radius as well as height and then choose the maximum step height and slope angle. **NOTE:** If the AI gets stuck inside objects this usually means the radius is not big enough. You also may be getting an error saying "NavMesh agent needs to be placed on NavMesh". This means your height or radius are completely wrong. Make sure to check them. Also we would suggest to create multiple profiles for each AI you have and that way each AI will have a unique behaviour. When you are done in here just go back to the "NavMesh Agent". Assign accurate radius and height once again (inaccruate values here can also result in the AI getting stuck) and you should be good to go! It is also worth mentioning that adding a collider to the AI is a good idea. That way it won't be able to go trough the player.

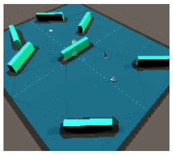


Agent Types

Humanoid

Name	Humanoid
Radius	0.75
Height	2
Step Height	0.75
Max Slope	45

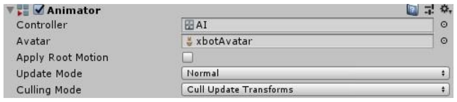
You should be now done. The pathfinding is being baked by the system at real-time depending on what the render distance is set to. **NOTE:** You won't be able to see the current calculated areas when in **EDIT** mode but once you play the game and select the pathfinding object everything will be visualised if you have selected the "Pathfinding" object. ***Pro tip:*** If you want an AI to move between multiple NavMesh Surfaces (for example floors) in a Scene, they must be



connected using a NavMesh Link. This can be done by adding the “NavMesh Link” component to the “Pathfinding” object. All that is left is to assign a start and an end point on it and a width.

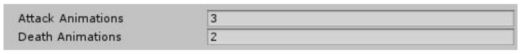
Setting up Animations, Sounds & VFX

Go to the animator and assign an avatar and an animation controller. We've provided both but you can use your own if you wish. Next open the animator controller and fill in the number of animations for attack depending on the chosen avatar/controller (there are 3 attack animations for the controller already included). Do the same thing for the death animations. As you can see in the controller, we've provided 3 attack animations there (remember the 0 also counts). **NOTE:** For every attack and death animation you will need a new trigger (e.g., if you have 9 animations, you should have 9 triggers). Every trigger should start with a capital letter and then you need to add the number. Your death trigger must look like this: Death0 and the attack ones – Attack4.

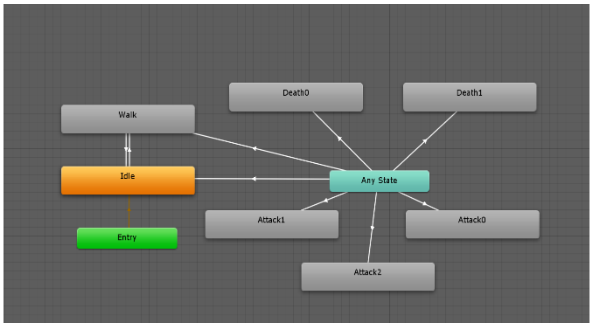


☑ **Animator**

Controller	AI
Avatar	xbotAvatar
Apply Root Motion	□
Update Mode	Normal
Culling Mode	Cull Update Transforms

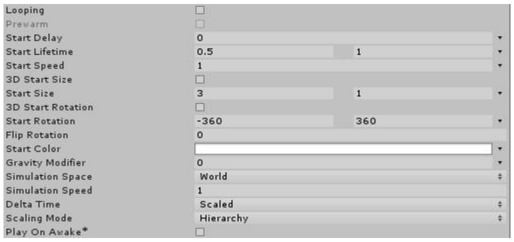


Attack Animations	3
Death Animations	2



Next you can add some VFX. Simply drag and drop the effects you want to be played to the corresponding array (e.g., if you want to play an explosion particle, you should put it under the "Attack Particles"). Just make sure you have all the particles inside the AI object.

You may also need to disable the "Looping" and the "Play on Awake" settings (don't forget to do that for their children if there are any as well!). Disable them if you need the particles to stop looping and not play when the game starts. The last step is to drag and drop the sounds you want to be played to the corresponding array (just like you did to the particles).

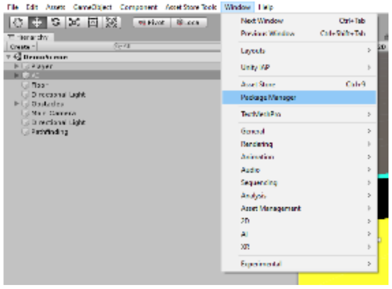


Looping	□	
Prewarm	□	
Start Delay	0	
Start Lifetime	0.5	1
Start Speed	1	
3D Start Size	□	
Start Size	3	1
3D Start Rotation	□	
Start Rotation	-360	360
Flip Rotation	0	
Start Color		
Gravity Modifier	0	
Simulation Space	World	
Simulation Speed	1	
Delta Time	Scaled	
Scaling Mode	Hierarchy	
Play On Awake*	□	

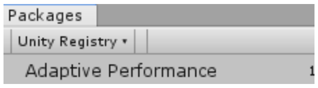
Next you can see that there is a setting called "Use Ragdolls". Leave it for now and we will be back on it in the RAGDOLL section. For now you can take a look at the INVERSE KINEMATICS section.

Setting up Inverse Kinematics (IK)

Time to set the inverse kinematics for our AI. This will make it realistically rotate its head based on what's happening around it. For this to work you will need to have Unity 2019.1 or higher and install the "Animation Rigging" package, following these instructions:



Go to: Window --> Package Manager --> Unity Registry



Then search for “Animation Rigging”:

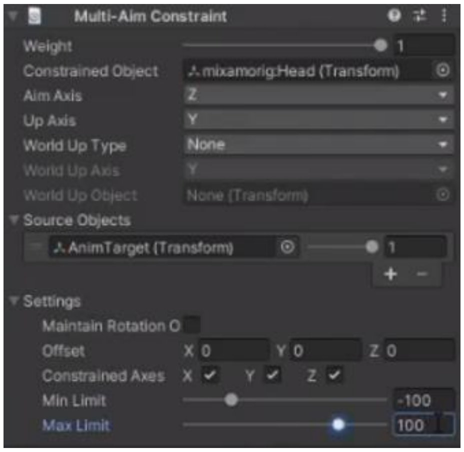
Animation Rigging

And insall it. Once the installing process is done you will have to select the AI object and click on:

Animation Rigging --> Rig Setup

Animation Rigging --> Bone Renderer Setup

A new game object called “Rig 1” will appear inside of the AI. Inside the “Rig 1” make another object and call it “HeadAim”. Then add the “Multi-Aim Constraint” component to it. You will then have to make another object called “AnimTarget” and position in to be a child of the “HeadAim” object and then drag it to where it says “Source Objects”. Next find your “Head Bone” and drag it into “Constrained Object”. Last but not least you should set a limit for the rotation of the head (in order to have a more realistic head movements of the AI and make it impossible for it to rotate its head on 180°and see what is behind it without moving its body) **NOTE:** If the head doesn’t rotate on the right axis this means that your object’s bones are inverted. You can fix this by changing the “Aim Axis”. Then grab the “AnimTarget” object and put it in the “IKPoint” field in the “Ultimate AI” script (on your AI object).



Multi-Aim Constraint

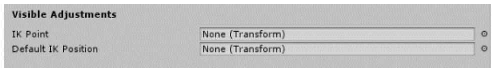
Weight	1
Constrained Object	mixamorig:Head (Transform)
Aim Axis	Z
Up Axis	Y
World Up Type	None
World Up Axis	Y
World Up Object	None (Transform)

Source Objects

AnimTarget (Transform)	1

Settings

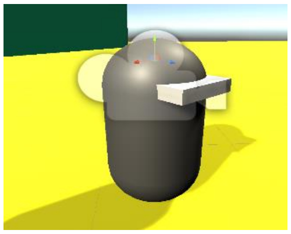
Maintain Rotation O	☐		
Offset	X 0	Y 0	Z 0
Constrained Axes	X ✓	Y ✓	Z ✓
Min Limit	-100		
Max Limit	100		



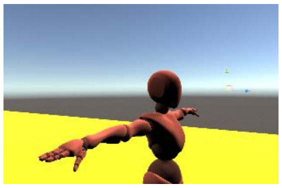
Visible Adjustments

IK Point	None (Transform)
Default IK Position	None (Transform)

Now it is time to set the “Positions”. In order to create them you’ll need 2 more empty objects – one inside the AI and one in the player. The player’s aim parent has to be positioned inside the head as shown:

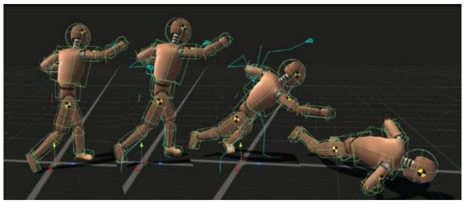


and the AI’s aim parent – in front of the head:



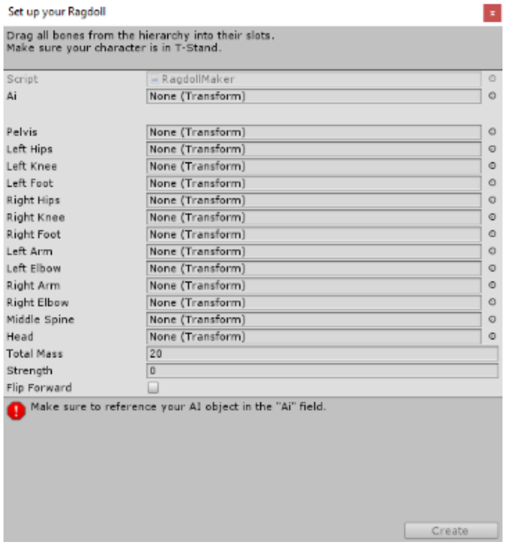
Once you get them positioned correctly just assign the parents inside the “Ultimate AI System” script. The player parent should go to the “Player IK Position” in the “Ultimate AI” script on the AI and the AI’s parent – inside “Default IK Position” in the “PlayerHealt” script attached to the player. Now you can go to the RAGDOLLS section.

Setting up Ragdolls



Let’s have a look at the ragdoll setting. If you are tired of animations and want unique results every time your AI dies this is totally what you are looking for. What are ragdolls? Ragdoll physics is a type of procedural animation, which is often used as a replacement for traditional static death animations. A ragdoll is basically a collection of multiple Rigidbodies (each of which is ordinarily tied to a bone of your AI character) tied together by a system of constraints that restrict how the bones may move relative to each other. Those constraints prevent the limbs for example from rotating unrealistically. When the character dies their body begins to collapse to the ground, honouring these restrictions on each of the joints' motion, which often looks more realistic and funnier actually.

So, time to create one ourselves. It is actually fairly simple. First, we need to turn on the “Use Ragdolls” setting. Once we have done that you will see that a “Build Ragdoll” button has appeared in the inspector. Hit that button and you should see this wizard open:



Set up your Ragdoll

Drag all bones from the hierarchy into their slots.
Make sure your character is in T-Stand.

Script	RagdollMaker
Ai	None (Transform)
Pelvis	None (Transform)
Left Hips	None (Transform)
Left Knee	None (Transform)
Left Foot	None (Transform)
Right Hips	None (Transform)
Right Knee	None (Transform)
Right Foot	None (Transform)
Left Arm	None (Transform)
Left Elbow	None (Transform)
Right Arm	None (Transform)
Right Elbow	None (Transform)
Middle Spine	None (Transform)
Head	None (Transform)
Total Mass	20
Strength	0
Flip Forward	□

Make sure to reference your AI object in the "Ai" field.

In the “Ai” field make sure to reference you AI object and right away you should see your bones automatically set them up. If you don’t’ see it, make sure your AI object has an “Animator” component and has a valid avatar. However there could be problems with your model so just to be sure check to see if all bones were set correctly. You can now adjust the mass of you ragdoll character and the strength. Then simply hit “Create” and your ragdoll is ready. You can now open the “Physics Debugger” window by going to Window --> Analysis --> Physics Debugger. Then click on “Collision Geometry” and right away every collider your character has will be visualized. Now you can tweak those colliders as they are not 100% accurate with every model. So for example if you are using our model – the “xbot” everything but the head collider is set properly. Because it is too small the results will not be very good. Therefore we can go to the head bone and resize the head collider to perfectly fit the head geometry. I think everything else is looking fine but you can always play around and improve a little those colliders and even the character joints or even add your own. ***Pro tip:*** in the physics debugger you should see 2 capsule colliders on both hands, 2 capsule colliders on both legs, 2 box colliders on the body and 1 sphere collider on the head – if any of these is missing this means it is too small. Make sure to scale them up to perfectly fit the corresponding body part. So there we have it – ragdolls. Next time our AI dies instead of playing a static animation its bones will use the laws of physics to realistically make our character die. **NOTE:** You can always reset your ragdoll by clicking the “Reset Ragdoll” button. Now the last thing you may consider doing is setting up HEARING.

Setting up Hearing

So, your AI is pretty much ready but it doesn’t have hearing yet? Well time to take care of that. First thing we are going to do is mark the “React To Sounds”. This will make more settings visible but let’s quickly get back to the first one – namely the ranges. You should now see a new range field – “Hearing Range”. This is how far the AI can hear. Now head back to where you previously were. You now have to choose – should your AI not change its path when it has detected and is going to an already heard point (whenever the AI hears a sound it picks a “heard point” a random point near the source of the sound it heard with accuracy based on the distance to it) or should it always change its pathing when it hears a new sound regardless if it is currently going to an already hear point. Make sure you choose one of those two options and based on your decision toggle either the “Lock To Heard Sound” variable or “Always Go To Heard Point”. All that is left now is to decide whether you want your AI to have realistic hearing and leave all the options below or make it depend on “luck”. If you want it to always hear as if it was a real character you are done here. Otherwise, you can pick one of the following options – “Go To Heard Point Randomly” which will ask you for a percentage and based on it there is a chance it will hear a sound or toggle the “Go To Heard Point After Multiple Sounds” which will ask you for an integer this time. This is how many sounds are required for your AI to hear before picking a heard point. And that’s it. You now have hearing.

Have fun using our AI system!

NStudios