Mecanim AI System Documentation

How to setup your custom enemy?

To setup the enemy you need to drag and drop the AIRuntimeController.cs script onto your prefab, or you can use the Add Component button from the inspector.

The AIRuntimeController script will take care to add all needed components to get the system running.

The required components are:

Animator

NavMeshAgent

AIRuntimeController

You can create and add any „Animator Controller“ to the animator component. Please follow the unity tutorials to get started on this.

Adjust the radius and height of the NavMeshAgent so it fits your prefab. Youdo not need to setup the speed nor the angular speed, because this be changed at runtime anyway inside the AIRuntimeController.

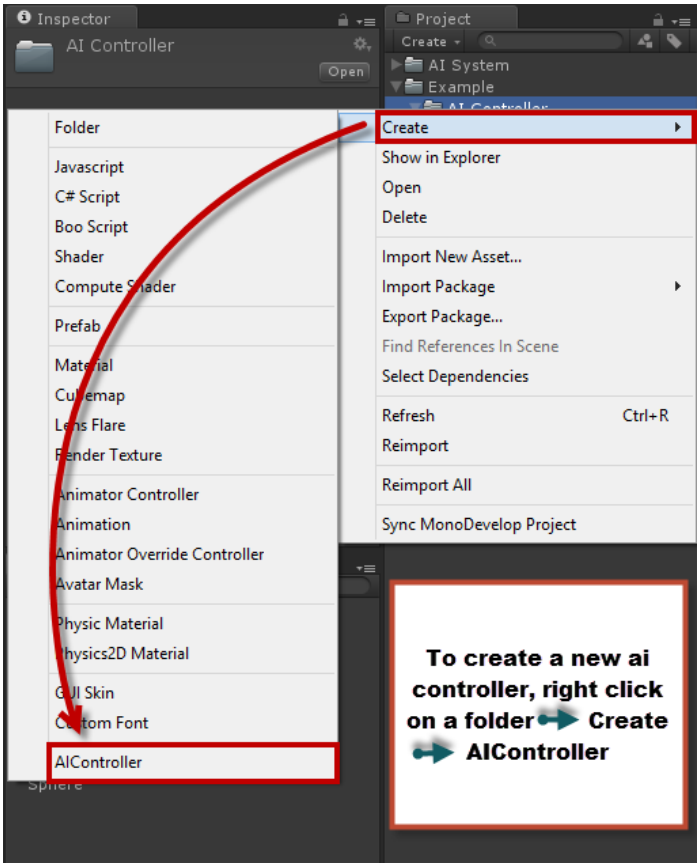
The AIRuntimeController has 2 variables that you can change in the inspector.

The first one „Controller“ is the base of the whole system. We will go in detail on how to create this controller in the second section.

The second variable is the level of your enemy. I am going to explain how the level influences the enemy when we go over the general setting in the properties of the AIController.

Please do not forget to bake a NavMesh for your scene!

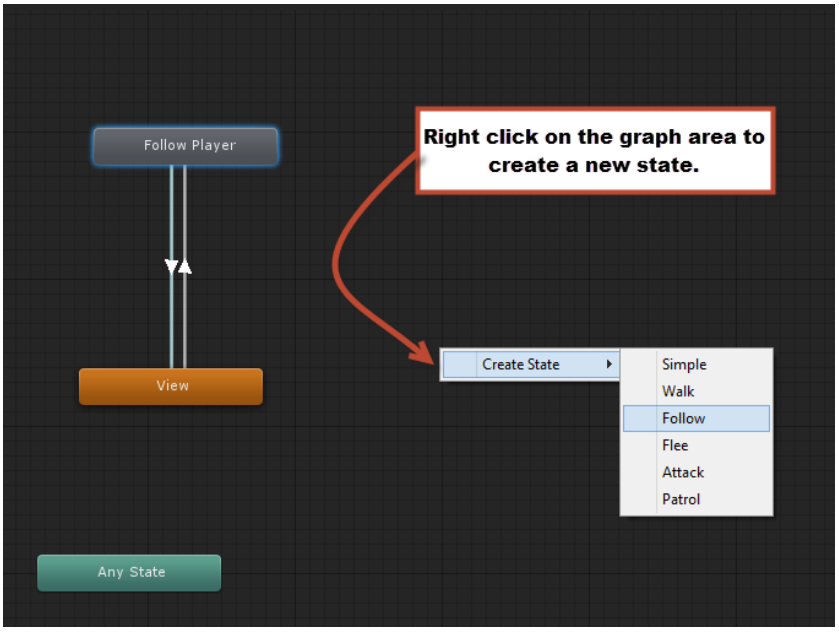
How to create a new AIController?



The creation of a new AIController works the same way as you create an Animator Controller.

Just right click in the Project window, go to Create ➡ AIController.

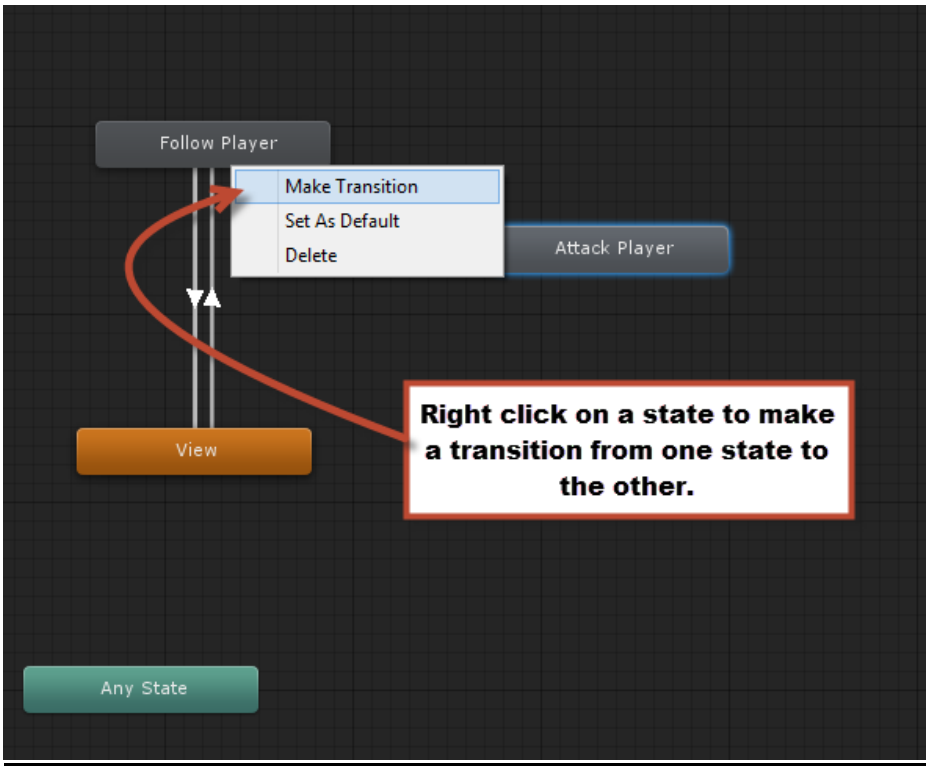
How to create new states?



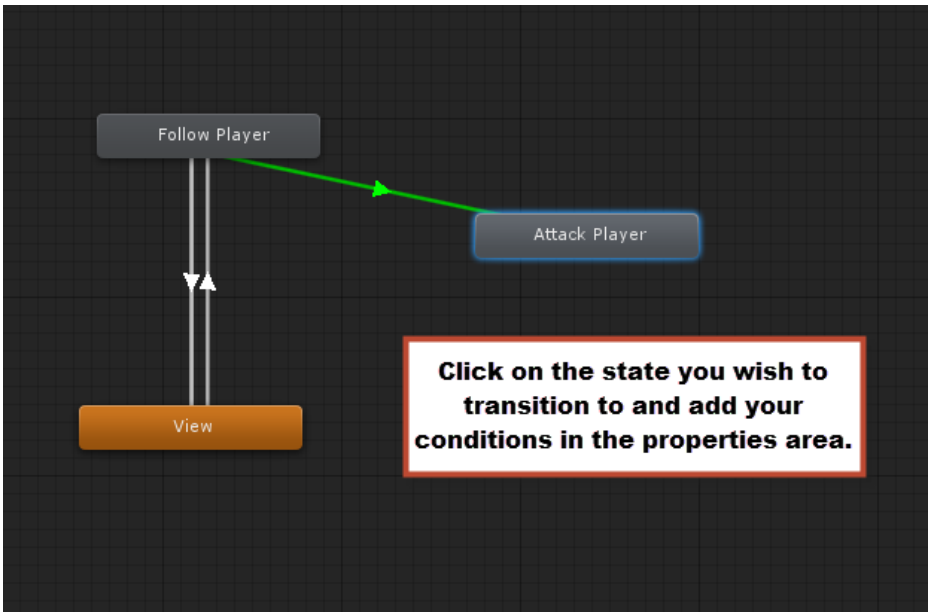
To create new states just click on an empty area in the graph, a dropdown menu will popup, where you can create new states.

How to link states?

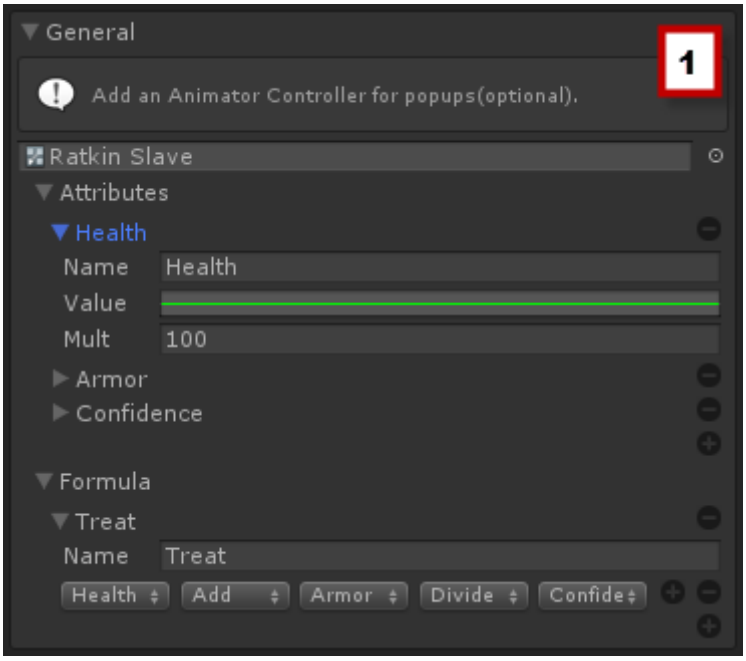
To make a transition from one state to another, you can right click on a state where you want to transition from, choose „Make Transition“ from the menu.



Afterwards click on a state you want to transition to.



Properties



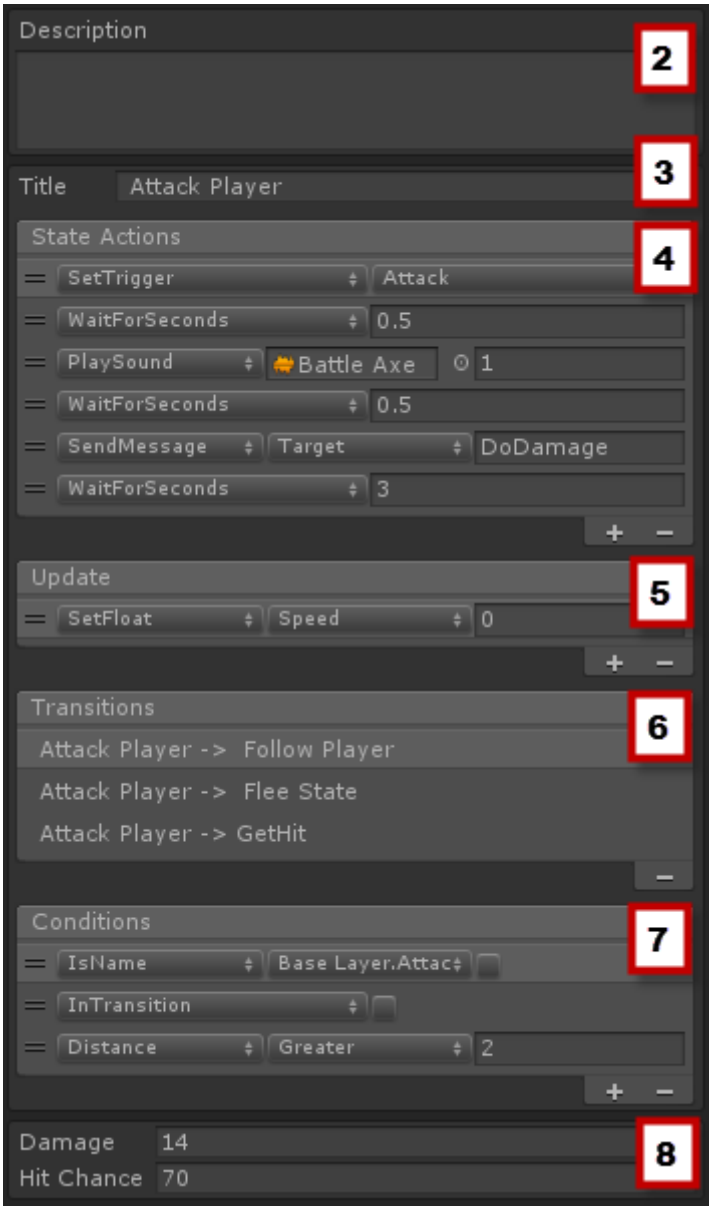
1. General information: In this area you can add attributes for your enemy. The enemy level influences the attribute max value. The time axis is the level x 0.01, the attribute value is the y axis value multiplied by the „Mult“ value.

   Example: If the level is set to 20 and the Mult value is 100 like in the picture on the left side(Cuve is the default linear one), the attribute value will be 20.
   You can also add a formula based on attributes and level and use it in the conditions.

2. The description is just for the developer and optional. But it is good to add one, so you can remember what you have done, when you reopen the controller.

3. Title ist he window title of the state, you can also get a state by this title, if you choose unique titles from the AIRuntimeController. If there are 2 or more states with the same title, the call of AIRuntimeController.GetStateByName(string title) will return the first one.

4. State actions are running in a Coroutine. This way you have a great control over timed actions. You can also influence the



performance by adding a WaitForSeconds action. State actions contain also Animator Actions.

5. Update actions run in void Update and are the same as state actions, however the WaitForSeconds action is ignored.

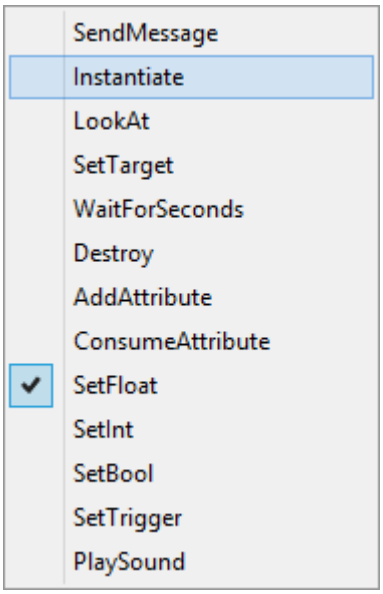
6. In this area are the transitions of the state diplayed. You can also break a transition by clicking on the minus button.

7. In the conditions area, you can set conditions when a state should transition to another.

State actions in detail.

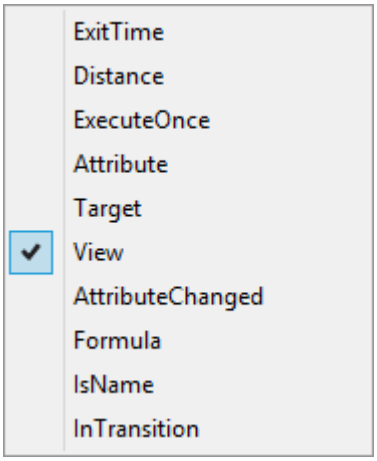
You can trigger state actions one by one. The big pro of a state action ist hat you can influence the time, when a state action should be triggered.

In state actions you can also use the Animator actions to set a float, bool, int or trigger parameter that you have defined in the Animator Controller.



- SendMessage: You can send a custom message with the unity messaging system to your self or to your target. This gives you the possibility to implement custom code that interact with state actions.
- Instantiate: You can instantiate a prefab at the target + offset position or self + offset position and of course at the world position.
- LookAt: This action works the same way as transform.LookAt() however the y axis is ingored and you can set to look at target or world position.
- SetTarget: With this action you can set the target of your agent by entering the tag of the game object. Distance, View and Target conditions are refering to this target.
- WaitForSeconds: Wait for a few seconds before you continue to execute your state actions.
- Destroy: You can set if you want to destroy the agent(Self) or the target. The extra field is optional for the time (after how many seconds the game object should be destroyed).
- AddAttribute: You can add a value to the current attribute value with this state. (CurrentValue+=value).
- ConsumeAttribute: Substract a value from the current attribute value.
- SetFloat(Animator Action): Sets the float parameter with the key and value.
- SetInt(Animator Action): Sets the in value by a string key.
- SetBool(Animator Action): Set the bool value added in the animator controller by key.
- SetTrigger(Animator Action): Set the trigger oft he animator controller. Same as Animator.SetTrigger(string key)
- PlaySound: Play an audio clip.

Conditions in detail



Add conditions when the AIController should transition from one state to another.

- ExitTime: After this time there will be a transition to the connected state.
- Distance: Check if the target is within or beyond distance.
- ExecuteOnce: No checks required the ai controller will execute the internal state once.
- Attribute: With this condition you can check the percentage of an attribute and behave if it is greater or less x.
- Target: Check if the target is equal null or not.
- View: Check if the target is in view field defined by angle and layer mask.
- AttributeChanged: Just a callback when an attribute changes.
- Forumula: Check the formula, which is set in the general settings.
- IsName: Check if the current AnimatorStateInfo is name…
- InTransition: Check if the animator is in transition.

How to add „and“ / „or“ conditions?

To add an „and“ condition just click on the plus button in the properties area. For „or“ conditions you need to make a new transition to the same state.

How to create a custom Action/Condition?

To create a custom action/condition you need to extend the CustomAction/CustomCondition class. You can derive from it and override the method Execute/Validate.

In this method you have the AIRuntimeController, Animator and State/AIRuntimeController as parameters and so a full controll over the owner of the AIController. CustomAction/CustomCondition is derived from ScriptableObject, so you will need to create an .asset file and assign it in the action/condition list in the AI editor choosing the type Custom.

To create this asset you have to right click on your created script, then create Action/Condition and save it somewhere in your project.

Contact Information

Konstantin Janson

Email: konstantin.janson@freenet.de

Skype: tharon211

Unity Forum: Zerano

Credits

The models included in this package are from Mr. Necturus. If you like them check out his site.

Furthermore i want to thank Kalamona who lets me use some of her particle systems. So if you like them search for the key word „Kalamona“ in the asset store and get more particle system for a very great price.

Video Tutorials

- How to get started?