RV HonorAI documentation

Join Ronis Vision's Discord support channel

note: always use online documentation if possible, as it will be more current than the pdf version provided with the package.

Online documentation

Video tutorials

Update note: because HonorAI uses SmartAI framework, always make sure to update SmartAI to the newest version **before** updating HonorAI to assure compatibility.

General overview

Honor AI is a utility-based solution for easy creation of various AI entities like monsters, animals, followers, villagers, hunters etc, with just a few clicks!
Thanks to the utility system it's based on, AI makes decisions based on stats and environment rather than hardcoded conditions, which makes AI fully dynamic.
Everything is designed and optimised to be usable in real game, so you can expect top quality, flexibility and performance.

Glossary

- Ai agent - the most general type of AI;
  any entity that uses SmartAI framework to handle it’s behaviour
- Character - core component of HonorAI AI agent.
  It can also mean more generally character as more specific type of AI agent

Working with HonorAI

Basics

HonorAI is an AI system built using SmartAI framework - please refer to SmartAI's documentation to get to know how to work with graphs, graph elements, how to debug AI behaviour at runtime etc.
While it's not mandatory to know how to work with SmartAI it's still highly recommended as it will help you understand the whole ecosystem and allow you to further customize and change AI for your project specific needs.
Click here to go to SmartAi documentation

Quick start guide

This quick start guide will show you how to setup working AI step by step.

- Create new scene.
- Add HonorAiManager object by clicking on toolbar *RVHonorAI -> Create HonorAi manager*
- Open Character creation wizard from the same menu
- Assign your character 3d model into *Character model* field
- Press *Create animator controller* and save it somewhere, it's good practice to have it in the same folder as your character 3d model
- Press *Create* button at the bottom of wizard window
- As we want our agent to move we have to create some ground for it and bake navigation mesh. Create a plane, place it at 0,0,0 and set scale to 10,1,10. This will be our ground.
- Set the plane as navigation static and bake navigation by pressing the Bake in Bake menu of Unity's navigation window. This will allow your agent to move on the plane.
- Congratulations! You just configured your first character.

Character component

Character component is core of HonorAI, all other systems and components are developed to support it or add additional features and helpers to get you up and running as quickly as possible.
It is designed to be as generic AI agent as possible, making very little assumptions about it's usage while still providing as much commonly needed features, like movement, combat, spatial awareness, groups and their relationship etc.

Character component has a convenient GUI divided in tabs that aims to centralize all AI and character related settings:

- General
- Movement
- Combat
- Animations
- Sounds
- Events
- AI

Most fields have hints so you can hover with a mouse over them to get their description.

Core character components

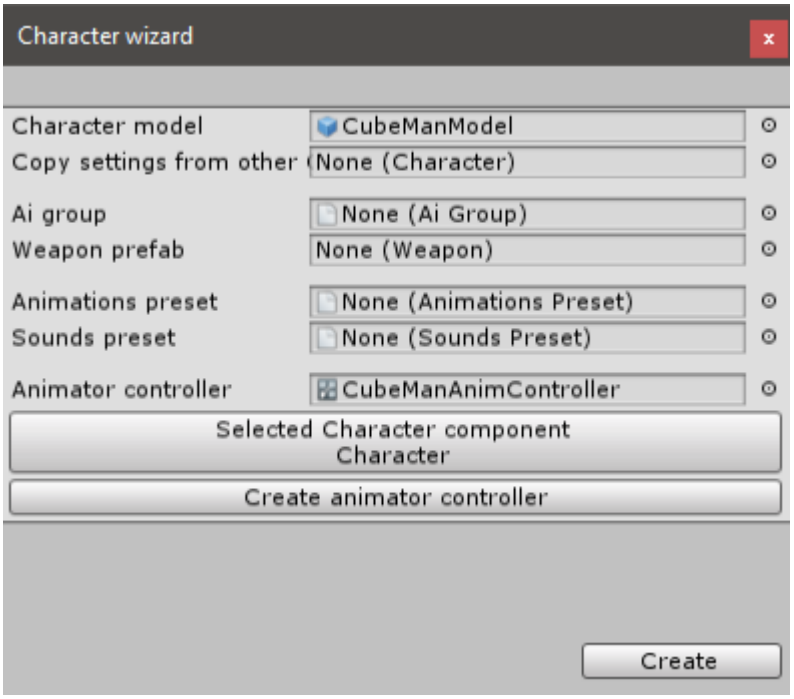
HonorAI Character is made with four main components that handle all AI features. All of them are designed to be loosely coupled, can be inherited and even completely replaced independently by user's own components that implements their main interfaces - named in parenthesis

- **Character(ICharacter)**
  handles the most of basic non-strictly AI related stuff like health, audio, events, gui and references
- **CharacterAi(ICharacterAi)**
  stores AI related stuff like list of targets, waypoints, nearby objects, AIGroup; it's mostly data storage class, there's not much logic other than looking logic
- **CharacterAnimation(ICharacterAnimation)**
  handles all stuff related to animations
- **UnityNavMeshMovement(IMovement)**
  handles unity's NavMeshAgent

Creating new Character

To create a new AI agent use character creation wizard.
Open character creation wizard by selecting from Unity's top toolbar ***RVHonorAI -> Open character creation wizard***.
After assigning necessary fields click *Create* at the bottom right corner of the wizard window to create your new character.
If you didn't assign some necessary fields or assigned not proper objects into fields you will get warning information telling you what's wrong and the create button will be greyed out.

Character creation wizard overview



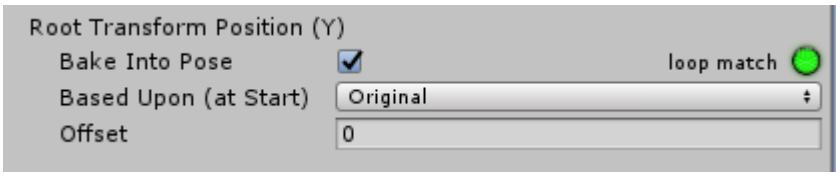
- **Character model**
  Here you assign a 3d model of your character. It need to have Animator component added on it
- **Copy settings from other**
  Here you can assign another character you setup earlier to use his settings. This will copy all variables like health, animations etc to new character
- **Ai group**
  Ai group that will be assigned for this character
- **Weapon prefab**
  If you want your character to have a weapon you can assign one here. This will automatically instantiate it in your character hierarchy and assign it

- **Animations preset**
  Automatically set all animations from provided preset
- **Sounds preset**
  Automatically set sound preset
- **Animator controller**
  Animator controller that will be automatically assigned to your character after creation
- **Selected Character component**
  Click to show all components that implement ICharacter interface;
  This gives you an option to select which implementation of ICharacter component you want to use - in future you'll be able to also select other of the core Character components
- **Create animator controller**
  Click it if you want to create new animator controller for your character and it will be automatically assigned

Character animations

- todo
- how to setup animations
- how to setup attack event
- explain difference in working with root motion vs non
- describe single animations
- .
- General info
    - Honor AI supports both humanoid and non-humanoid(generic) animation types, and both root motion(RM) and in-place animation.
      It is important to remember not to mix them, especially moving ones.
    - For humanoids you can use RM animations and still use them in non-RM configuration
    - RM vs non-RM setup
        - These two setups differ fundamentally in a way they move Character:
          with non-RM setup all movement is done by Unity's NavMeshAgent;
          with RM setup all movement is done by animation system and so speed of movement of your character will be defined by animations and their playback speed
- Preparing animations

- Import options
  - it’s good idea to set “Root transform position (Y)”
    “Bake into pose” to true



  - For all non-moving animations like attacks or dying you should set “Bake into pose” under “Root transform position (XY) to true
-

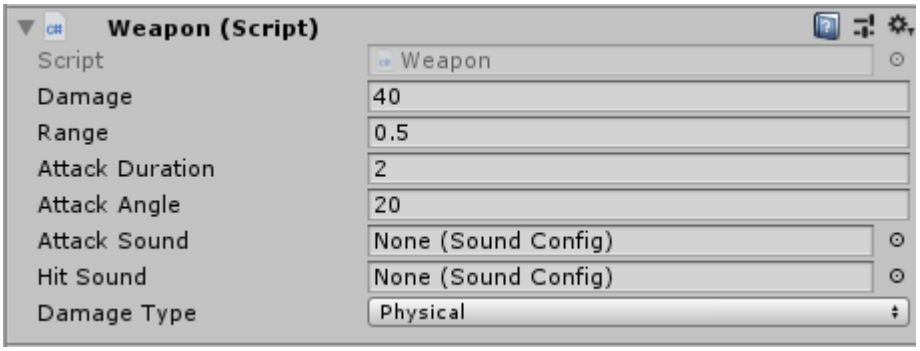
Ai groups and relationship

HonorAI features groups that allow you to easily setup relationships between different characters and groups of characters.
To create a new AiGroup click RMB in the *project window -> Create -> Honor AI -> Ai group*.
To setup relationship between groups simply select one of them and drag and drop the other group onto a relevant field - Enemies or Allies.
Enemies will immediately attack each other. Allies are used in evaluation of winning chance of fight.
If characters from different groups aren’t added as enemies or allies they are neutral and so won’t engage and will simply be ignored in fights.
Note that you can also set specific characters to treat neutral characters as enemies in Character inspector.

Weapons system

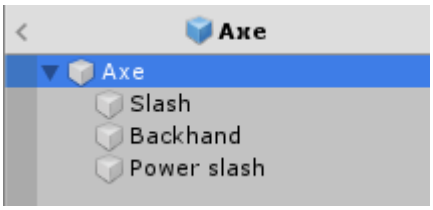
HonorAI uses a reusable weapons/attacks system that allows you to define the weapon and it's attack as separate game objects and prefab it.
Each character can have many weapons and each weapon can have many attacks.
To create a new weapon simply create a new game object and add a Weapon or DistanceWeapon component to it.
Such system allows to easily share attacks between different characters, eg. you can create a weapon prefab "melee" with punches and kicks animations and add it to all your characters, that way you can have every character in your game in addition to his normal attacks have the same set of melee combat attacks.

Weapon component:

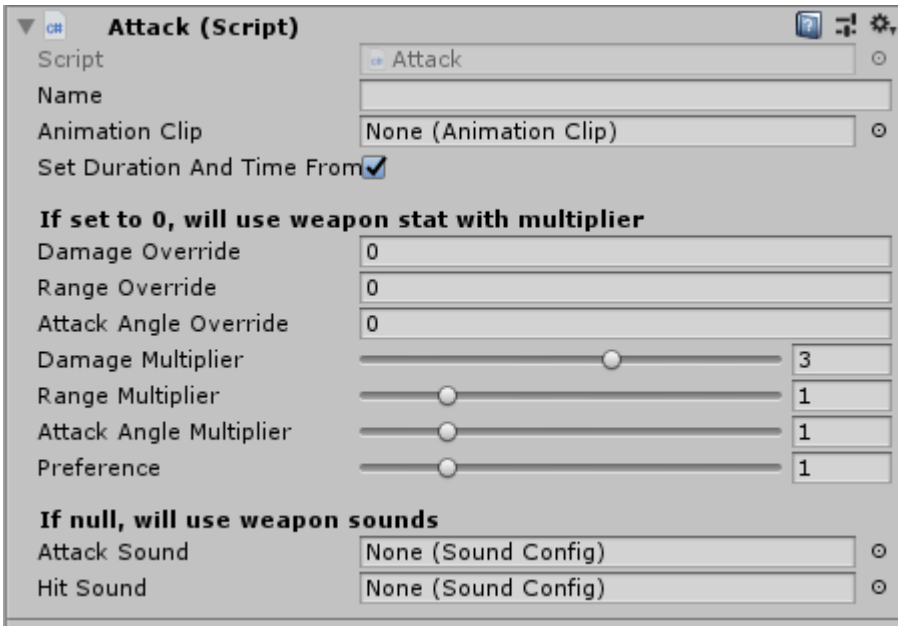


The simplest setup is to just add one weapon. Attack will be added automatically at runtime. With such setup, attack animation will be selected randomly.
To add attack to weapon simply add Attack component.
Attack components can be added on the same game object as a weapon or as his children.
Adding attacks allows you to set animation for each attack and change the chance for selecting it based on "preference" value.

Example weapon setup:



Attack component:



Each attack has a set of overrides and multipliers that allows you to modify base weapon stats or override them. Overrides work if they're set to value other than 0 and multiplier value is then ignored, otherwise multiplier value is used.
Eg: weapon have a damage value of 50, and have a power attack that will be slow but will deal more damage, with damage multiplier set to 3 it will have a damage value of 150.
Make sure that the animation clip you selected for attack is added in his Attacking Animations array under Animations tab of Character component.
If the animation clip isn't set for attack, it will be selected randomly.

Setting up ragdolls

- Create separate prefab using your character's model
- Use unity's ragdoll wizard to add rigidbodies, joints and colliders to it
- Assign ragdoll prefab onto ragdoll slot under 'General' tab of Character inspector
- Optionally add DisableRigidbodiesOnSleep component on ragdoll root game object and configure it accordingly to your needs

Debugging Character

To check targets our Character knows about, check the “Target infos” list in the CharacterAI inspector. Similarly, all nearby objects scanned by scanner will be in the “Nearby Objects” list.
All of those essential information will be integrated under the “AI” tab of Character inspector for more centralized debugging experience in future updates.

Character spawners

Character spawner is a component that allows you to simply configure agent spawner.
Spawner can invoke spawn automatically on scene start and can also be called by API and via Unity events system.
It also has a list of waypoints you can configure that will be automatically added to all spawned characters.

Ai zones

AiZone is a component that allows you to invoke some Unity event when a player or other object enters its trigger.
Zones can also be coupled with spawners so that zone will invoke spawning, but also can activate and deactivate all characters spawned by coupled spawner, effectively creating simple spatial optimization for your game - AI will not waste precious cpu time when they are of of player range/visibility.
You can configure which tags will enable and disable AIs in the zone from the inspector.

Modifying graphs

If you want to make some change in AI that is not possible using just inspector configuration you will need to modify provided AI graphs. To do so, make a prefab variant of BaseGraph if you want to make sure your graph will be updated with new versions of HonorAI. You can also just copy BaseGraph and use it if you don’t mind getting out of sync with future updates.
Note that BaseGraph has nodes referencing other graphs like CombatGraph, you have to make your own versions of each one of those graphs(only if you want to

modify them), whether you copy or make variants of them is up to you, just like with BaseGraph.

Presets

You can create, export and import presets for both animations and sounds.
To create a preset click RMB in Project view to open the context menu, then go Create -> HonorAI and select Sound or Animation preset. Such presets can be then imported to any character using his inspector under their respective tab.

Exposing own fields in Character inspector

You can expose any(serialized by Unity) field from your own components added on your character hierarchy(it has to be on the same GO as the Character component or child of it).
To do so, make your component implement the IExposeCharInspectorFields interface. Then you can choose to display all serialized fields of your component by returning true in the ExposeAllFieldsToCharInspector property.
DefaultCharInspectorTab is a property where you can decide under which inspector tab all fields will be shown.
Another option is to expose only fields you want to - just use CharacterInspectorFieldAttribute on fields you want to expose, where you can also specify under which tab field should be shown.

Integrations

GameKitController

- Installation
    - GKC integration files are located in the GameKitController.zip file in the Integrations folder. Simply extract all those files into your GKC folder.
- Enabling interactions between both assets
    - For your HonorAI characters add the “GKCIntegration” component. This will also automatically add the necessary “CharacterFactionManager” component - configure it accordingly to GKC documentation - “Faction System” field has to be assigned.
    - For your GKC AI characters add the “HonorAIIntegration” component and assign all fields.
- Note that both assets have their own relationship systems. You need to configure both of them in a consistent way to make sure that they all will recognize each other as enemies, otherwise it will result in a situation where AI from one asset attacks the other but not vice-versa.

Content overview

Examples

HonorAI has several example scenes to help you get started.
Example scenes are located at ***Assets\RVModules\RVHonorAI\Examples\Scenes.***

Troubleshooting/Faq

- Characters don’t move or don't move correctly
    - Make sure you have setup scanners layers properly and that terrain(floor) can be scanned
- Characters don’t attack each other
    - First thing to check is if they should attack each other in the first place - easiest way to do this is to remove assigned AI groups and tick “treat neutral characters as enemies”
    - If they still don’t attack each other it’s probably because they don’t know about their surroundings! Make sure they both have colliders added on the same game object as Character component and that their environment scanners have Scanner layer mask set to layer your characters are set to
- I don't know what my characters are doing/why are they doing
    - Select your character at runtime and press the Debug AI button on their Character component. Refer to SmartAI manual to check on how to debug AI graphs
- My character disappears when it's killed
    - If you don’t see any errors that means you checked the “Use ragdoll” field in your Character inspector but didn’t assign any actual ragdoll to create on their death. Assign ragdoll or uncheck “Use ragdoll” to use death animations instead
- I see errors in console when enter play mode/when characters attack each other
    - Make sure you have added HonorAiManager object to the scene
- I want to change how feature X works
    - Find out which component implements the feature you want to change, create your own inheriting from it, override said method and replace mentioned component with your own. Get more info about provided components here.
      If you have problems with overriding, or some member is not virtual and you think it should be, let me know on Discord, I always make sure that you can integrate AI into your game without modifying any of RV source code