# NPC Framework Documentation

## Version: 0.1.dev

## General Description

This is a brief introduction to the NPC Framework. This text introduces the basic concepts, explains the main set up, usage and how to extends the framework and usage examples.

## Introduction

The NPC framework is a controller-centralized set of components to manage every aspect of humanoid agents within games and simulations in Unity. These characters can be player controlled or not. Some of these aspects are: perception radius, IK, path finding, steering, obstacles detection, social forces, behaviors and animations, among other agent services. The most important aspect of the framework is its capacity to be fully extendable and customizable based on anyone's needs. These can be done in two simple ways:

1. Parameterizing the basic built in services.

2. Via the implementation of plug n' play modules using provided interfaces.

Finally, the framework provides many extendable components, such as an Animator Controller, which can also be fully extended based on development needs.

On a higher level, the implemention is mainly divided the following way:

NPCController - single point of interface for all the agent's components.

NPCPerception

NPCBody

NPCAI

.. modules and subcomponents

These are some of the basic provided services:

Components Basic Services

----------- --------------

**PERCEPTION** Perception Radius and View Angle

Perceived objects / agents.

**BODY** Affordances: basic atomic actions

i.e. Go To, Look At, Orient Towards, etc...

Social forces steering

Basic IK

Forward obstacle detection

**AI** Pathfinding

As you can see, each category offers a simple layer of services to make a character operational. More complex agents would extend this capabilities with further development.

As mentioned, the implementation of the framework is modular, which means that this can be extended with any component which implements some provided interfaces. These are:

INPCModule

IPerceivable

...

On top of these interfaces, some subcomponents can be used to extend also the behavior and interaction ability of agents with any objects of a scene. More to come with usage examples.

## Setup

0) You will need to add the NPC code to the by either importing the provided unity package or adding the code manually.

The code is distributed among two folders (relative to Unity's Assets folder)

/Scripts/NPC/\* code, and the ...

/Editor/NPC/NPCController\_Editor.cs

Now, for any humanoid agent, simply:

1) Add the "NPCDefaultAnimatorController" Animator Controller from /NPC/NPC Agent/Animations Controllers to the agent.

2) Drag the NPCController script from the Project View and drop it on the Inspector of the intended empty agent's model.

As any other empty basic model, before hand, this will only have the Unity's basic "Transform" settings and an Animator Controller.

The NPCController will initialize many components (most will be hidden in the inspector) automatically. Some of these are: Rigid body, a capsule collider (if none has been added to the agent already - for example a Ragdoll model), a NavMeshAgent (used for automatic basic path resolution) and other components.

From this point on, you can manage almost any aspect from the NPCController component of the agent.

3) Adjust the agent's parameters. That's it. These parameters are (top-to-bottom order):