Full NPC
System Documentation

Table of Contents:

1) Introduction

The NPC System is a system designed for Unity 3D games.

2) Installing the NPC System

Note: You can skip theses setup steps if you go to prefab and place the NPCSystem GameObject into your scene.

- *Step 1: Setting Up the NPCSystem Object*
  1. Create an empty object named NPCSystem in your scene.
  2. Add the NPCSpawner component to this empty object. Most fields in the script are self-explanatory, but detailed documentation is available at the end of this document.
  3. Add all your NPC prefabs to the Base NPC Prefab list and assign their animation state controller to the NPCAnimatorStateController field.

- *Step 2: Configuring the Spawnpoint and Waypoint Managers*
  1. Create an empty child object of NPCSystem and name it SpawnpointManager.
  2. Add the SpawnpointManager component to this object.
  3. Repeat the process with another empty child of NPCSystem named WaypointsManager, and add the WaypointManager script instead of SpawnpointManager.

- *Step 3: Using the System*
  1. Refer to the "Using the System" section for detailed instructions.

3) Using the NPC System

The system is easy to understand. To add spawnpoints (where NPCs will spawn), select the SpawnpointManager object and hold the Left Shift key while right-clicking to create a spawnpoint at the desired location in the scene. To add waypoints, follow the same process. It is recommended to duplicate the WaypointsManager if you want to have multiple paths for NPCs. Avoid using a single WaypointsManager for all paths, as this could cause NPCs to move unpredictably.

4) Common Issues

When adding waypoints to the scene, the editor scripts (waypointEditor and spawnpointEditor) must be in the same folder as other MonoBehaviour scripts. However, if you want to create a build of your game, you must move these scripts into the Editor folder. If you don't, your build will fail.

5) NPC Spawner Component Reference

Here's a brief explanation of the fields in the NPCSpawner script:

- NPCAnimatorStateController (RuntimeAnimatorController): The animation state controller for all NPCs. Create one with the animations you want and assign it here.
- BaseNpcPrefabs (List): A list of all available NPC prefabs, which will be chosen randomly before spawning.
- NPCCount (Int): The number of NPCs to spawn.
- RespawnCheckInterval (Int): The number of seconds before checking if NPCs are dead and whether they should disappear and respawn.
- PlayerProximityRadius (Float): The radius of the sphere within which we check if the player is present. If not, NPCs can respawn.
- Min NPC Speed (Float): The minimum speed NPCs can have.
- Max NPC Speed (Float): The maximum speed NPCs can have.

6) Features Added in Updates

Version 1.0:

1. **NPC Damage and Death:**
   All NPCs can be killed using the TakeDamage method from the ShootableNPC script, which is attached by default to every NPC. Upon death, NPCs transition into a ragdoll state, simulating realistic physical behavior during their death phase.
2. **Static NPCs:**
   Static NPCs have been added to the system. To create a static NPC, simply attach the StaticNPCConfigurator script to the NPC GameObject you want to be static. You can assign a role to the NPC, such as a mechanic or a doctor. (Note: This feature is still under development, so further improvements are expected in future updates.)
3. **Fleeing Behavior:**
   A fleeing behavior has been added to all NPCs. When a player fires a gun, NPCs will flee in the opposite direction of the player. This behavior is triggered when your firearm script calls the following method:
   GunshotEventManager.GunshotFired(rayHit.point);
   This allows NPCs to detect gunfire and react accordingly, adding more immersion to your game.