

# Project Resurgence - NPC/Enemy AI System Reference

Rev 5/29/2018

This document is a run down of the NPC/AI scripts currently in use in Project Resurgence. It should be referenced in tandem with the AI\_System\_Layout.pdf document located in the project.

**NOTE: Variables, initializations via Start()/Awake() and strictly internal optimization methods are not directly referenced in this document, this is by design - please see script headings for current information on these information.**

**NOTE: All scripts will be finalized to this document as part of Alpha launching as stated in HnP.**

<b>Project Resurgence - NPC/Enemy AI System - Rev 5/29/2018</b>	<b>1</b>
AI_Enemy_Charger_Targeter.cs	1
AI_NPC_Assistor_Nav_Point.cs	1
AI_Enemy_Debug_Signal_Attack.cs	1
AI_Player_Helper.cs	2
AI_Debug_Draw_Zone.cs	2
AI_Enemy_Assistor_Controller.cs	2
AI_General_Navpoint_Pool.cs	2
AI_Wake_Up_Zone.cs	2
AI_NPC_Shotpoint.cs	2
AI_NPC_Assistor_Nav_Mover.cs	2
AI_NPC_Navigator.cs	3
AI_Spawner.cs	3
AI_Enemy_Controller.cs	3
AI_Enemy_Assistor.cs	4
AI_Enemy_Charger.cs	4
AI_NPC_Targeter.cs	4

- `AI_Energy_Charger_Targeter.cs`
  - `OnTriggerEnter()`
    - If a Player or NPC then attack sequences are initiated (if not holding)
  - `Hold_For_New_Target_CO()`
    - This timer controls the holding timer for attack changes.
  
- `AI_NPC_Assistor_Nav_Point.cs`
  - This script manages the NPCs claim and holding of nav points.
  
- `AI_Energy_Debug_Signal_Attack.cs`
  - **NOTE: This is a debug script that signals when attacks are triggered - this is where attacks will need to go (tied to a coroutine timer).**
  
- `AI_Player_Helper.cs`
  - `Check_If_Moving()`
    - Checks if the player is moving or not and sets `isMoving` bool to true/false
  - `OnTriggerEnter()`
    - Sets the player zone variable to the entered zone - via event
  - `OnTriggerExit()`
    - Sets the player zone variable to null on exit - via event
  
- `AI_Debug_Draw_Zone.cs`
  - This script simply draws zones for use in the editor.
  
- `AI_Energy_Assistor_Controller.cs`
  - This script is not active in the current build. It rotates the assistor group and can be used to randomize the appearance of enemy attacks.
  
- `AI_General_Navpoint_Pool.cs`
  - `Claim_Open_Navpoint()`
    - This is called publicly and checks for open nav points and then return the point to the caller.
  - `Release_All_Navpoints()`
    - This releases all nav points owners.
  - `OnTriggerEnter()`
    - This scrambles and releases all nav points when the player enters.

- **NOTE: This may be causing an issue with the null exception issue - this would be a good place to look.**
  - OnTriggerExit()
    - This releases all nav points when the player exits.
  - Mini\_Scramble\_Nav\_Points()
    - This scrambles the nav points so they will be varied on each entry.
- AI\_Wake\_Up\_Zone.cs
  - **NOTE: This is a test script, it can be used but should be replaced by a functional spawner connected to an object pool.**
  - OnTriggerEnter()
    - When the player (by tag) hits a zone - the game objects in the list connected to the zone will activate.
- AI\_NPC\_Shotpoint.cs
  - **NOTE: This script is in need of optimization - it is a test script only.**
  - StartShooting() and StopShooting() initiate and deactivate ShootingCycle\_CO coroutine
  - ShootingCycle\_CO() is the actual shooting cycle
  - ShootingCycle\_CO() is the individual shots
  - The OnDrawGizmos simply draws a locator box in the editor if debug is active.
  - Shooting and stop shooting are publicly called
- AI\_NPC\_Assistor\_Nav\_Mover.cs
  - This controls the locating of nav points from the player for the NPCs.
  - Reset\_Nav\_Points\_CO()
    - This cycles the raycast check via a coroutine.
  - Reset\_Nav\_Points()
    - This resets the nav points based on a ray cast call.
- AI\_NPC\_Navigator.cs
  - This script is listening for events. If you are not familiar with C# events and delegates please review the documentation for C# and Unity.
  - Update()
    - This checks if there is a shot and then shoots.
    - **NOTE: This should be removed from update and optimized into a coroutine.**
  - GotoPoint()
    - This sets the nav agent destination to a point.

- SetPoint\_Player()
    - This requests a nav point assistor from the player and sets it as the goto\_this target. NOTE: There an occasional null exception that occurs here on the line #107 - it will need to be fixed.
  - SetPoint\_Zone()
    - This requests a nav point assistor from a zone if the player is in one. There is a null exception that occurs here on line #125 - it will need to be fixed.
  - Turn\_To\_Closest\_Energy()
    - This uses a LookAt() method to turn the NPC towards the enemy.
  - Shoot\_If\_Shot\_Exists()
    - This checks if there is a raycast hit and then engages shooting if it exists.
  - Goto\_CO()
    - This is the goto nav cycle coroutine.
  - Check\_If\_Player\_Moving\_CO()
    - Coroutine check for if the player is moving or not.
  - Ease\_Out\_Stop\_Moving()
    - Coroutine stops the NPC movement is the player stops moving.
- AI\_Spawner.cs
    - NOT YET IMPLEMENTED
- AI\_Energy\_Controller.cs
    - Goto\_Target()
      - Set the nav agent target to the target variable position.
    - Attack\_Target()
      - NOT YET IMPLEMENTED - may not be required
    - Get\_Assistor\_Head()
      - This gets the head of the assistors (the main parent for assistors)
    - Get\_Assistor()
      - This collects all of the assistors from the assistor group into a list.
    - CO\_Goto\_Target
      - This cycles the command to goto the target.
- AI\_Energy\_Assistor.cs
    - The enemy assistors help enemies attack. When an enemy collides with one they will be routed to another.
    - OnTriggerStay()
      - Based on the assistor number a collision will route the the enemy to a new assistor. This helps with left and right attacks.

- **AI\_Energy\_Charger.cs**

- Goto\_Target()
  - Sets the nav agent destination to the target.
- Goto\_Target\_CO()
  - Cycles the goto target coroutine.
- Attack\_Using\_Assistor\_CO()
  - This cycles attacks using this coroutine.
- Check\_For\_Attack\_Targets()
  - If there is an attack target in the list then set it as the current target - if a target was set then cycle it to the end of the list for the next check.
- Get\_Attack\_Target\_Assistor()
  - This retrieves the attack targets assistors and adds them to the AI list.
- Get\_Closest\_Assistor()
  - This will get the closest assistor from the attack target (player or npc) and will return it.
- Engage\_Attack\_Using\_Assistor()
  - Switches from Goto\_Target\_CO coroutine to Attack\_Using\_Assistor\_CO coroutine.
- Attack\_Using\_Assitor()
  - If agent is not null (to prevent error - handler should be added) - Then the new destination is set for the nav agent.
- Advance\_Attack\_Pass()
  - This is not yet implemented but is designed to initiate a retreat after 4 attack passes.

- **AI\_NPC\_Targeter.cs**

- **NOTE: This script should be reworked - enemies in the project should be layered instead of using tags - then the tag checks can be removed.**
- OnTriggerEnter()
  - This adds objects that are tagged "Enemy" to a list that is referenced when picking closets targets.
- OnTriggerExit()
  - This removes objects that are tagged "Enemy" to a list that is referenced when picking closest targets.
- Check\_For\_Closest\_Enemy()
  - This cycles through the current target list, and checks the distances between them, and then sets the target variable to the closest one.
  - This check runs as a coroutine.