

Information on Optimized NPC:

Optimized NPC: The AI pawn used in Plastic Battlegrounds

About the Current NPC AI:

- Our existing AI soldier is built in BP as a pawn and does not use behavior trees, EQS, or any UE AI systems for the purpose of performance. Our AI soldiers are custom Blueprint pawns using AddActorWorldOffset and hand-authored logic for pathfinding, targeting, and animation syncing. They don't use Behavior Trees or UE's CharacterMovementComponent — instead relying on timers, linetraces, and world offsets for movement. This system must be made fully multiplayer-compatible with server authority, efficient replication, and consistent visuals across clients.

Code Breakdown of Existing Singleplayer NPC:

Event Graph: **Contains Movement logic (encapsulated in “Movement” comment):**

- **Restart Movement:** This is where the movement logic begins
 - Random speed is set at randomized time intervals
 - Sets “**MoveForward**” timer which actually moves the AI via the Incremental Movement Offset collapsed node
- **Incremental Offset Movement:** uses AddActorWorldOffset to move to location give by SetNav function, uses the first index of path points and updates to the next path point if reached (look at the next part of logic to see it updating path points)
- Includes logic for when the NPC is ragdolled but still alive
- **Make Pawn Walk On Surface:** NPC Pawn is kept to the ground via a linetrace that hits WorldStatic objects. If nothing is hit, the pawn starts falling and fall logic is activated in movement logic.
- AnimationBP movement variables: Movement speed variables used in the Animation BP are calculated at the end of the movement logic
- **Fire Weapon:** AI weapon firing logic
 - **Spawn Projectile:** Spawns a BP_TracerBeam projectile at target
 - To simplify, we may be using Hit Scan logic instead
 - Hit scan needs to not hit friendlies
- LOD: changes movement intervals depending on distance to player meaning NPCs further away will move at a lower timer interval and will use Actor Offset toward their destinations using larger distances (closer AI offset at much finer increments so player doesn't notice they're being offset).

- This may need to be modified to fit multiplayer such that it will change depending on distance to the player client instead of player pawn index 0.

Initialization Graph

Is run at BeginPlay

- AI adds self to team array found in game mode
- Starts movement logic with “Restart Movement”
- Command Init: ignore this, not used in MP yet
- AI is brought to the ground when spawned via a linetrace
- Init AI: Initializes AI (find AI Graph below)

Damage Graph: This is where the NPC takes damage, gets limbs dismembered, and dies

- There are
- Breaks into pieces via the Apply Gore Damage node which uses the UEGoreSystem
- Takes damage from Plastic Damage event and Radial Damage, which breaks their limbs off

AI Graph:

- **Set Nav:** Finds location to move to via “Find Path to Location Synchronously” node which gives array of locations.
 - Most of the logic in this graph has to do with moving to “capture point” objectives, or is split into different command responses, these are not relevant at this moment, only the parts determining movement to the “Target” are relevant since the AI only deal with enemies in MP and not objectives yet.
- **Set Target:** Finds closest enemy.
- **SingleVisibilityCheck:** Linetrace to closest target to see if they are in sight, if not then do AllVisibilityCheck (next)
- **AllVisibilityCheck** finds the nearest *visible* target if closest enemy is not determined to be visible from SingleVisibilityCheck
- **TargetVisibility** - Does line trace to target to determine if AI should be firing at the target or not (only fires at target if visible). This is specifically to determine if AI should fire, does not determine target to set as the previous visibility checks do
- Fire at Target logic initialized here
- SetRotation: Sets rotation of AI - either toward enemy if they exist or just the direction they’re walking
- SetPitch - Changes the units’ pitch up and down depending if target is above or below
- **Update Team Color** - changes team color of the AI based on **Team Int** variable, will set itself to be Green (0), Tan(1), Blue(2), or Grey(3). Green and Blue are allies and are enemies of the Tan and Grey who themselves are allies with each other.

NPC Physics Graph

This is where the NPC simulates physics when falling over and getting up

- Events such as Explosions can knock NPCs down
 - NPCs stand back up after being knocked down
- Normally if the player is at arms length of the NPC, they will start simulating physics while standing so the player can grab them or punch them. For the purpose of MP, we don't need to have them simulate physics initially

Command Graph: Currently not used in MP

Toy Soldier NPC Specifications/Tasks

Note: All systems are already made for singleplayer, but need replication and likely restructuring to work for MP

Allow NPC to move around and animate via ABP

- Move to location synchronously
- Uses Add movement offset toward location

NPC acquires target and faces toward them

NPC fires at target

- Shooting animation
- Muzzle flash if close

Dies and ragdolls and breaks apart. Other NPCs stop targeting them

About the game:

Plastic Battlegrounds released one year ago to critical success with nearly 8k ratings at 4.9 stars out of 5 and reaching #4 Top Seller on the Meta store due to its flexible tools that allow players to relive their childhood toy fantasy battles. Our team is a passionate group of developers aiming to bring back creativity to players' who want to simulate toy battlefields, and AI soldiers are a big part of making them immersive.

Plastic Battlegrounds on Meta :

<https://www.meta.com/en-gb/experiences/plastic-battlegrounds/7796723770415439/>

[Habie147 plays Plastic Battlegrounds](#) - breaks down the gameplay very clearly