3D Minimap System - Official Documentation

Created by: Manyak Studios **Support & Feedback:** help@manyakstudios.com

Welcome to the **3D Minimap System**. This highly optimized, Instanced Static Mesh (ISM) based system detects actors within a defined area and creates a reduced-scale 3D representation in front of the player.

---

1. Quick Start & Basic Setup

To add the 3D minimap to your game, simply drag and drop the `BP_MapItem` Blueprint into your level.

- By default, any actor within the scan area will be automatically scanned and added to the 3D map representation. You don't need to do anything extra for standard static actors!

---

2. The Tagging System (Managing Actors)

To keep performance high, the system uses specific Tags and Components to separate static, dynamic, and unwanted actors. All tagging operations are done by selecting the actor, searching for "Tags" in the Details panel, and adding the following tags to the array:

- **What we DON'T want to see (`IgnoreMap`):** If there are actors you do not want to include (e.g., environmental props like basic walls or floors), add the `IgnoreMap` actor tag. Actors with this tag will be completely ignored by the scanner.
- **Separating Movable Actors (`Map_Active`):** To track the real-time location of movable actors (rockets, vehicles, puzzle objects), you MUST add the `Map_Active` tag to *both* the actor and the component. This prevents the system from doing unnecessary math calculations on objects that remain stationary.
- **Tracking Spawned/Destroyed Actors:** If your game dynamically spawns or destroys enemies/items during gameplay, you need to add the `BP_MinimapTrackable` component to those specific actors so the map can register them on the fly.

---

3. Customizable Editor Parameters

When you select the minimap in your level, you can tweak the following parameters in the Details panel to fit your game's needs:

- **Scan Radius - 3D:** The physical area in the world to scan.
- **Static Map Size:** The physical size of the minimap itself.
- **Current/Min/Max Scale:** Controls the initial, smallest, and largest zoom levels of the map.
- **Rotation Speed - 3D:** Adjusts how fast the map rotates during examination.
- **Check PD:** Calculates the distance between the player and the map.
- **Tick Map:** The refresh rate of the minimap's calculations.
- **CheckRadius:** The timer for detecting when an actor enters or leaves the map's tracking distance.

---

4. Optimization & Performance Guide

We have optimized this system by separating active, passive, and dynamic actors wherever possible. Please note that tracking actors always has a performance cost. At a 0.04 tick rate on an empty map with 256 actors, the performance footprint remains very low and manageable for up to 256 concurrent actors. However, pushing beyond this 256-actor limit, or increasing the tick rate, causes the calculation cost to scale up exponentially, resulting in heavy FPS drops.

Built-in Optimization Features:

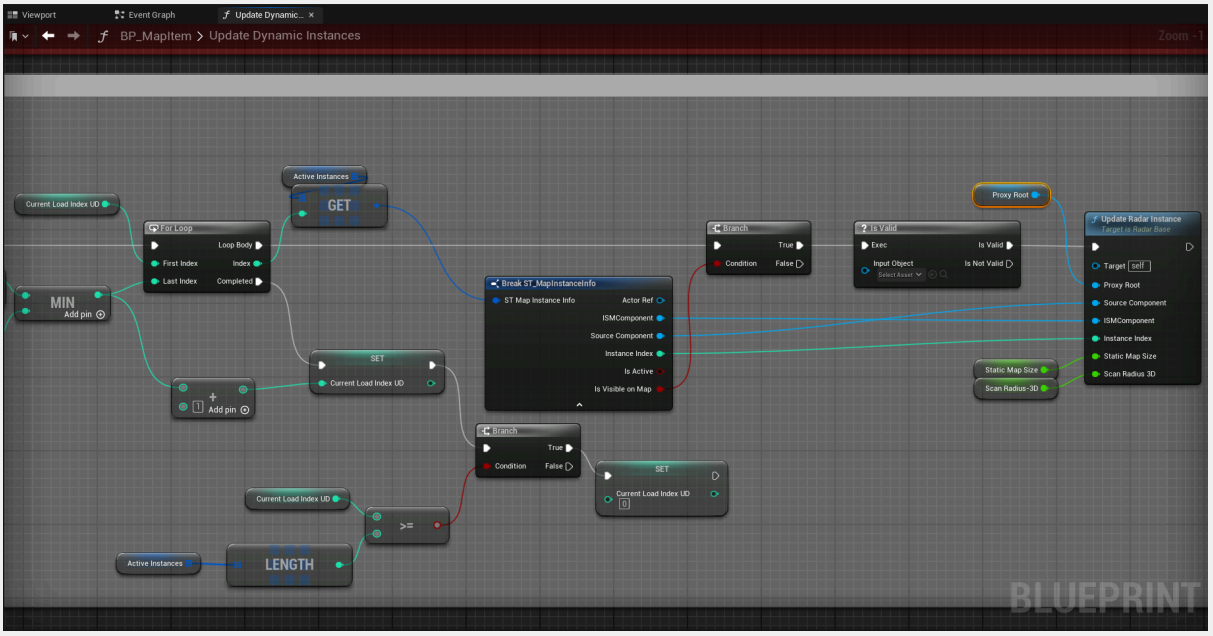
1. **Visibility Check:** If the minimap is not visible on the player's screen, the system completely stops running background math.
2. **Distance Culling:** As the character moves farther away from the minimap, its internal Tick Rate is automatically reduced to lower the CPU workload.
3. **Motion Masking:** While the player is in "Inspection Mode" (rotating the map), the background tracking FPS is slightly reduced to guarantee buttery-smooth rotation.

Developer Tips for Maximum FPS:

- Avoid displaying 256+ actively moving actors inside the radius at the exact same time.
- Do not overlap multiple 3D map systems inside the same working area.
- Lower the `MapTimerRadius` value so the system stops tracking actors at a closer distance.
- If you have massive swarms of enemies, **increase the `Tick Map` value**. The minimap's visual updates will be slightly less frequent, but your overall game FPS will improve massively

- **Visual Scale vs. Tick Rate:** Please be aware of the relationship between the physical size of your 3D minimap and its Tick Rate. If you scale the 3D map to be very large on the screen, a low tick rate will cause noticeable visual stuttering that catches the eye. Conversely, if you keep the map size smaller, you can safely lower the tick rate to save performance. At smaller scales, the reduced update frequency won't be nearly as disturbing or noticeable to the player..

⚡ **Optional C++ Performance Upgrade** For massive performance optimization, you can easily migrate the heavy radar math to C++. Open the `BP_MapItem` blueprint, go to `Update Dynamic Instances`, and copy the source code from the purple comment box at the bottom into a newly created Actor-based C++ class (e.g., `RadarBase`). Replace the `YOURPROJECT_API` macro in the header with your exact project name, then save and compile. Finally, return to the same blueprint function, keep the initial "Is Valid" check, delete the old math nodes, and connect the new `Update Radar Instance` node to run your minimap natively in C++ with maximum efficiency.



---

5. Blueprint Architecture & Asset Breakdown

For advanced users who want to modify the system, here is how the core logic is structured:

Core Logic (`BP_MapItem`):

- **Event Graph:** Trigger boxes start the system when the player enters the area and presses 'E'. Static actors are locked, timers begin, and the `StartExamine` logic allows mouse rotation and mouse-wheel zooming.

- **CheckPlayerDistance:** Reduces map tick rate based on distance and shuts down logic if out of sight.
- **Scan Actives:** Registers active actors separately. *(Note: Currently supports standard Static Meshes. Full ISM support for tracked entities is in experimental development).*
- **MM_CheckActorRadius:** The core bounds-checker. Routes to `CreateInstance` and `DeleteInstance` to add/remove dots as they enter or leave the zone.

Included Assets:

- **`BP_GreenScanner`:** Handles the 'E' key interaction setup, animation, and material.
- **Materials:** Includes `Material_Map` (featuring 7 different visual styles), `Material_Function` (fade-in animation), and `Scan_Material` (visualizes the scan area).
- **Audio:** Comes with 4 optimized sound assets and 1 sound configuration.
- **Test Maps:** The `Map_Test` folder includes pre-configured moving actors, spheres, and cylinders to test the tracking logic safely.

Important Performance Example: If you are designing a dense game level—for example, packing 256 active actors into a tightly confined 3500-unit radius and trying to track all of them on the minimap simultaneously—you will experience severe FPS drops. Please note that "256actor 0.04 tick" is not an exact mathematical limit, but the background calculation costs start reaching disruptive and uncomfortable levels once you surpass the 200-actor mark. Please adjust your map's tracking radius and actor density accordingly to maintain a healthy frame rate.