The Persistent Lightmaps Tool

Developed by SWeeb

This tool helps optimize lightmap usage in scenes with LOD groups by automatically copying baked lightmap data from LOD0 to all other LOD levels. It reduces bake times, avoids redundant baking, and can save memory at runtime. The tool includes a main tab for applying and removing lightmaps, a statistics tab to see what's included or excluded, and an about tab with version info and documentation links. Designed to work seamlessly on large scenes without requiring manual setup for each LODGroup.

Basic Overview:

Statistics Tab

This tab scans the current scene and shows useful data about all LODGroups found. It displays how many are included (using PersistentLightmap), how many are excluded, and how many are ineligible (missing static flag or only have LOD0). It helps you quickly understand which objects are affected and estimate potential memory savings.
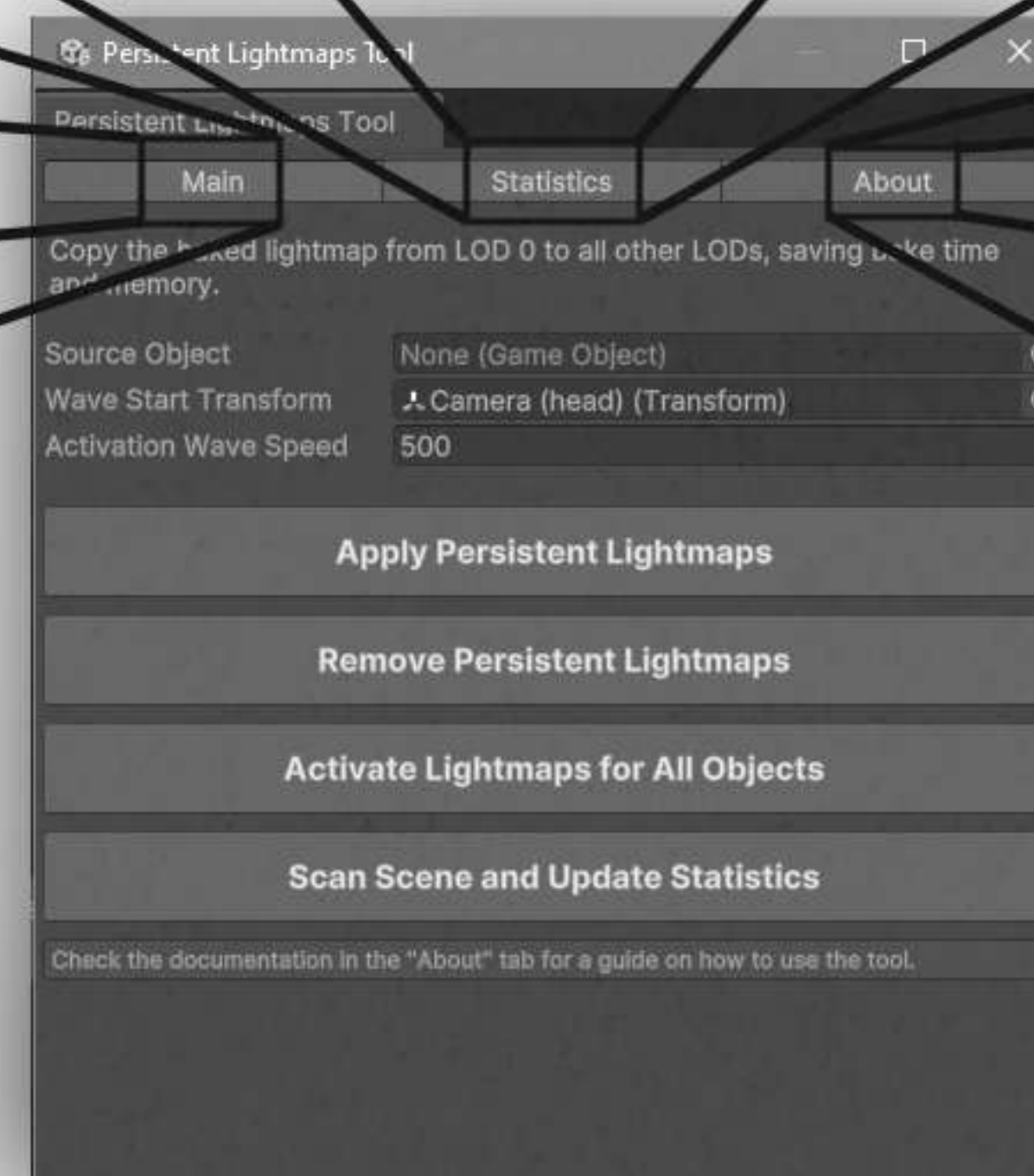
Main Tab

This tab is where you run the tool's main actions. You can apply or remove persistent lightmaps, activate them for all objects in the scene, and rescan to update statistics. It also lets you choose the source object and adjust settings like the activation wave start point and speed.

About Tab

This tab explains the purpose and scope of the tool. It describes how the tool helps transfer baked lightmaps from LOD0 to other LOD levels to reduce bake times and memory usage. It also provides version information, author credits, and a quick link to the official documentation for further details.

Persistent Lightmaps Tool

| Main | Statistics | About |

Copy the baked lightmap from LOD 0 to all other LODs, saving bake time and memory.

Source Object — None (Game Object)
Wave Start Transform — Camera (head) (Transform)
Activation Wave Speed — 500

Apply Persistent Lightmaps

Remove Persistent Lightmaps

Activate Lightmaps for All Objects

Scan Scene and Update Statistics

Check the documentation in the "About" tab for a guide on how to use the tool.

The Persistent Lightmaps Tool

Developed by SWeeb

The Main Tab is where you run the core actions of the tool. Here you can choose your Source Object and set optional parameters like Wave Start Transform and Activation Wave Speed. You can then apply persistent lightmaps to all LODs, remove them, activate them across the entire scene, or rescan to update statistics. Designed to keep the workflow fast, organized, and easy to control.

Main Tab:

Source Object:

The Source Object is the root GameObject that contains your LODGroups. The tool will recursively search through this object and all its children to find eligible LODGroups, copy the baked lightmap from LOD0, and apply it to other LOD levels. This lets you control exactly which part of your scene gets processed.

Wave Start Transform:

Defines the starting point of the activation wave when applying lightmaps at runtime. The tool uses this transform's position to calculate distances to each renderer and applies the lightmaps in a smooth wave effect, based on the distance and the configured wave speed. Useful for large scenes to avoid applying everything at once and create a more seamless activation.

Activation Wave Speed:

Controls how quickly the lightmap activation wave moves through the scene. A higher value means lightmaps are applied almost instantly to all objects, while a lower value makes the wave move more gradually from the Wave Start Transform, creating a smoother effect at runtime. Adjust based on scene size and desired visual impact.
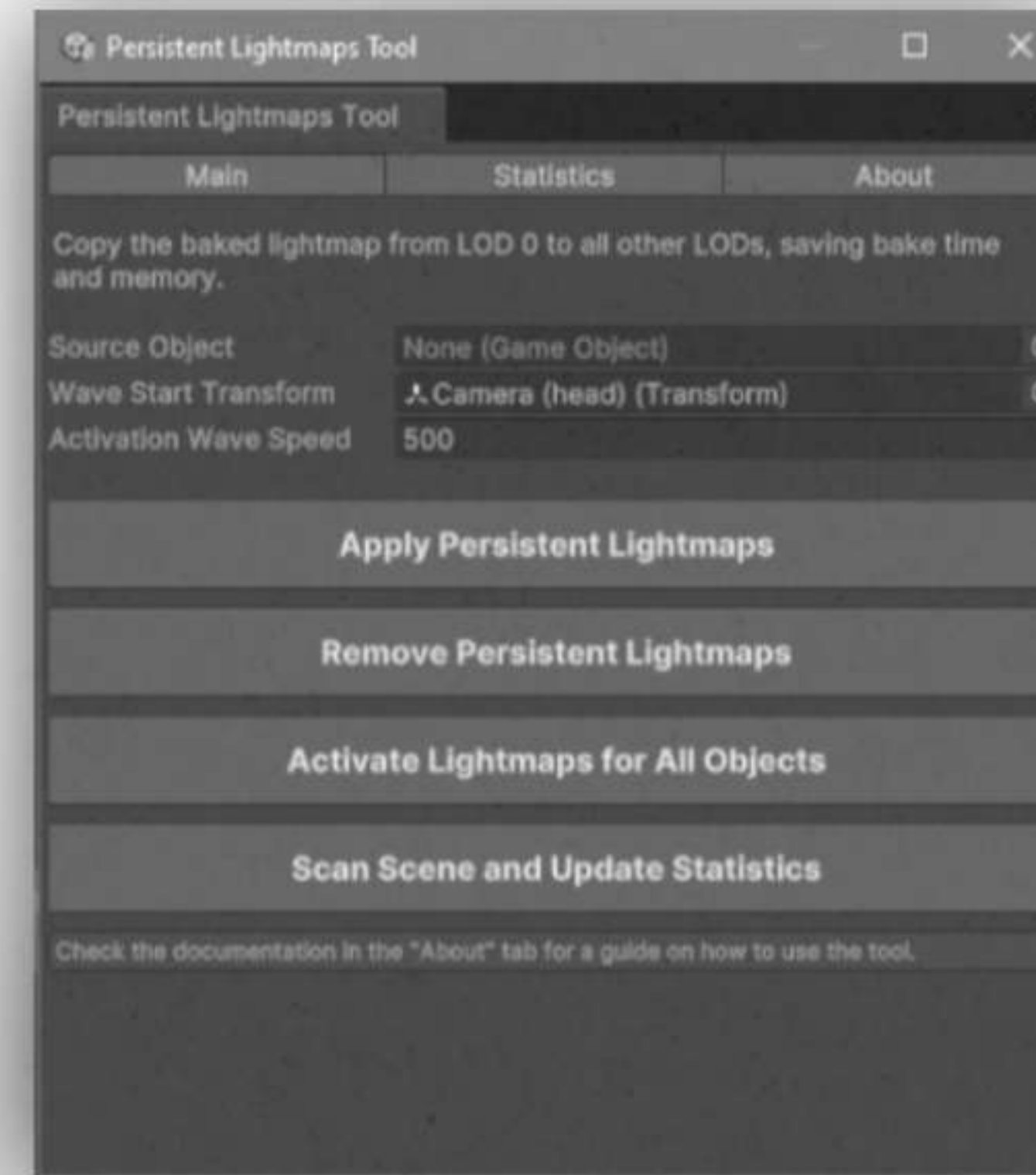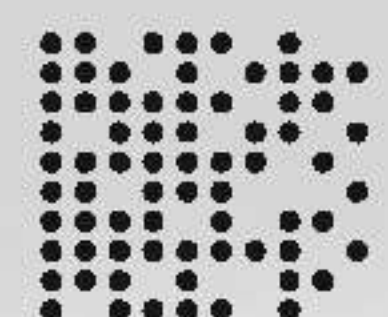

Apply Persistent Lightmaps:

This button copies the baked lightmap data from LOD 0 to all other LOD levels within the selected Source Object hierarchy. It recursively searches for eligible LODGroups, skips those marked as excluded or not static, and applies the lightmap settings to improve bake efficiency and reduce memory usage. After processing, it shows a summary dialog with the number of affected objects and optionally prints their names to the console.

Remove Persistent Lightmaps:

This button removes previously applied persistent lightmap data from all LOD levels within the selected Source Object. It resets lightmap indices and scale offsets to default values for eligible LODGroups, excluding those marked as excluded or non-static. After completion, it shows a summary dialog with the count of affected objects and can print their names in the console for review.

Activate Lightmaps for All Objects:

This button activates the saved persistent lightmaps on all objects in the scene that have the PersistentLightmap component. It immediately applies the stored lightmap data without delays or waves, useful for previewing the final lighting setup during development. Note that this action is for preview purposes and is not required for the tool's core functionality.
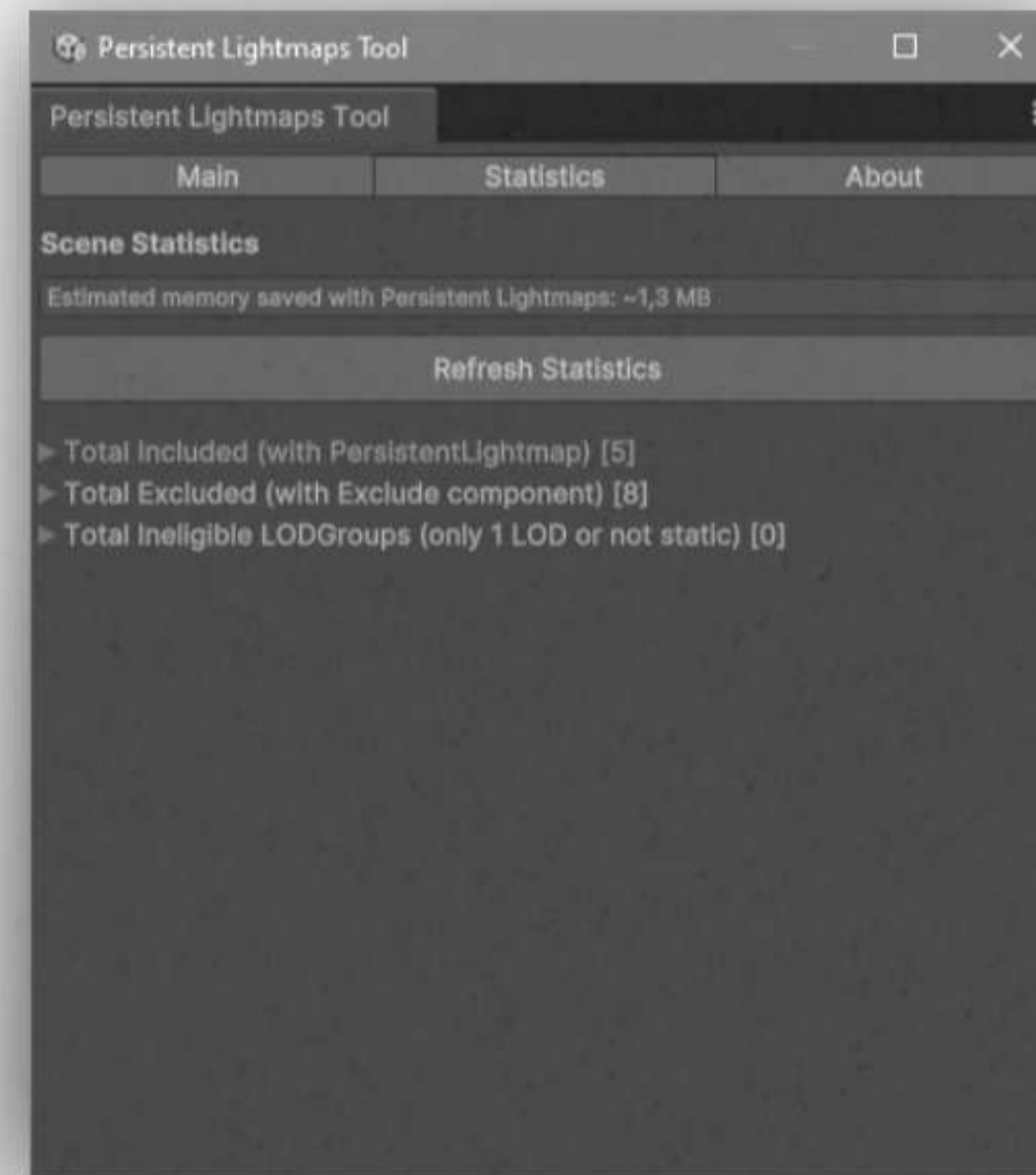
The Persistent Lightmaps Tool

Developed by SWeeb

This tab provides an overview of all LODGroups in the current scene. It shows counts for included LODGroups (those with PersistentLightmap applied), excluded ones (marked to skip), and ineligible groups (those with only one LOD level or not marked as static). Use this tab to quickly assess which objects are affected by the tool and identify potential optimization opportunities.

Statistics Tab:

Included:
Represents the number of LODGroups in the scene that have the PersistentLightmap component applied. These are the objects where baked lightmap data from LOD0 has been successfully copied to all other LOD levels. Counting only these ensures you know exactly how many objects are benefiting from persistent lightmap optimization.

Excluded:
Represents the number of LODGroups that are explicitly marked to be skipped by the tool using the ExcludeFromPersistentLightmap component. These objects will not have their lightmaps copied or modified, allowing you to exclude specific models from the persistent lightmap process.

Ineligible:
Counts the LODGroups that are not eligible for persistent lightmap application. This includes groups with only one LOD level (usually just LOD0) or those whose GameObjects are not marked as static. These objects cannot benefit from the tool's optimizations and are skipped during processing.
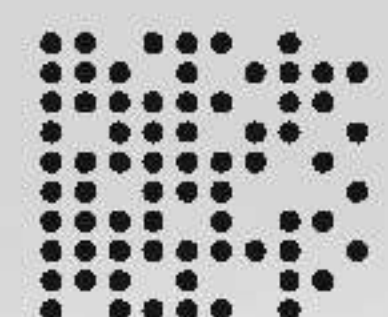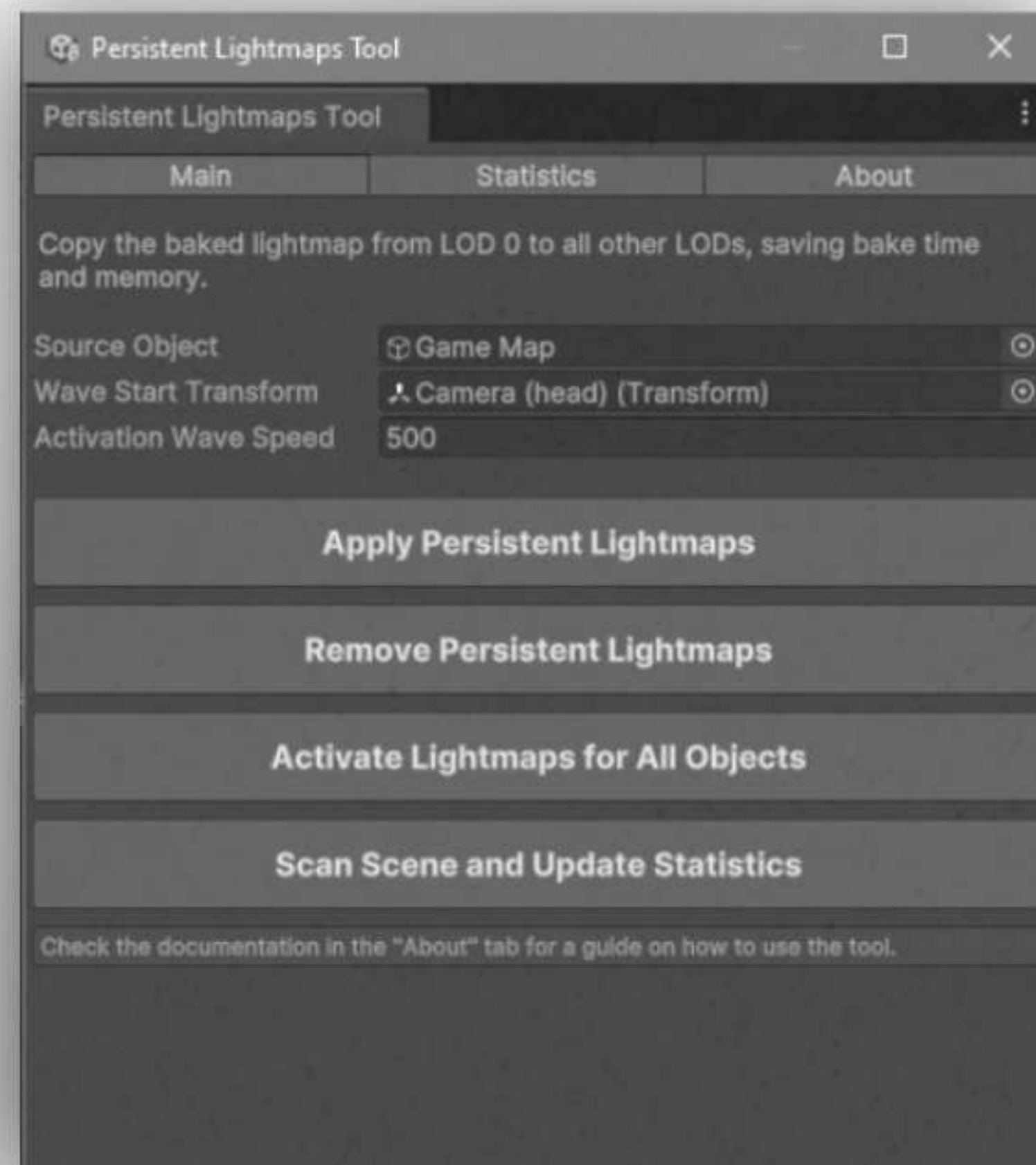
The Persistent Lightmaps Tool

Developed by SWeeb

Usage 1:

1.

Source Object: Pick the GameObject in your scene that acts as the root.
The tool will search recursively inside this object to find all LODGroups to process.

2.

Wave Start Transform: Choose a Transform as the starting point for the activation wave effect (useful for large scenes).
Activation Wave Speed: Adjust how fast the wave applies the lightmaps.
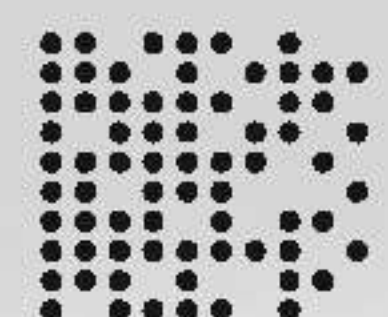Higher values = faster, lower values = smoother activation. (leave it at 500 for the best result)

3.

Press Apply Persistent Lightmaps.
The tool will copy the baked lightmap data from LOD0 to all other LODs of eligible objects (static, not excluded, and with multiple LOD levels).
A summary dialog shows how many objects were affected.

The Persistent Lightmaps Tool

Developed by SWeeb

Removing Persistent Lightmaps

Usage 2:

1.

Go to the Main tab of the tool.

2.

Make sure your Source Object is set correctly
(the parent object containing the LODGroups you want to clean).
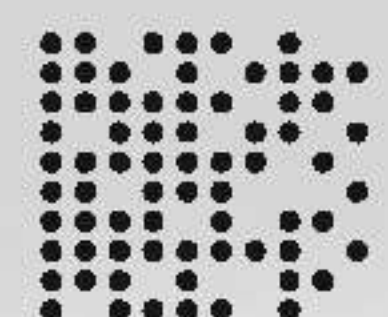
3.

Click the Remove Persistent Lightmaps button.

The tool will:
– Recursively search inside your Source Object.
– Find all eligible LODGroups (objects that aren't excluded and are marked as static).
– Remove the saved lightmap data by clearing lightmap indices and scale offsets beyond LOD0,
and removing the PersistentLightmap component if present.

At the end, you'll see a dialog showing how many objects were affected, and you can
optionally print their names to the console for review.

Note: This operation does not delete your baked lightmaps from disk,
it just resets the data on the selected objects, so they won't use the persistent lightmaps anymore.

Checking Tool Statistics

Usage 3:

1.

Go to the Statistics tab of the tool.

2.

Press the "Refresh Statistics" button.
This will scan your entire scene and gather data about every LODGroup.

3.

You'll see a foldout showing:
– Total Included: Number of LODGroups that have the PersistentLightmap component and are ready to use persistent lightmaps.

– Total Excluded: Number of LODGroups marked with the ExcludeFromPersistentLightmap component to skip them.

– Total Ineligible: Number of LODGroups that can't use persistent lightmaps because they only have one LOD level or aren't marked as static.
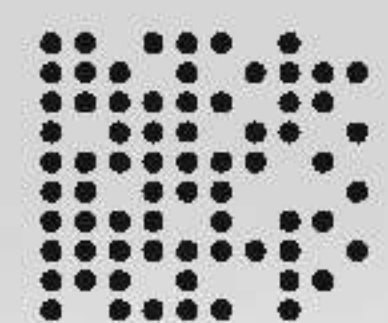
The Persistent Lightmaps Tool

Developed by SWeeb

Usage 4:
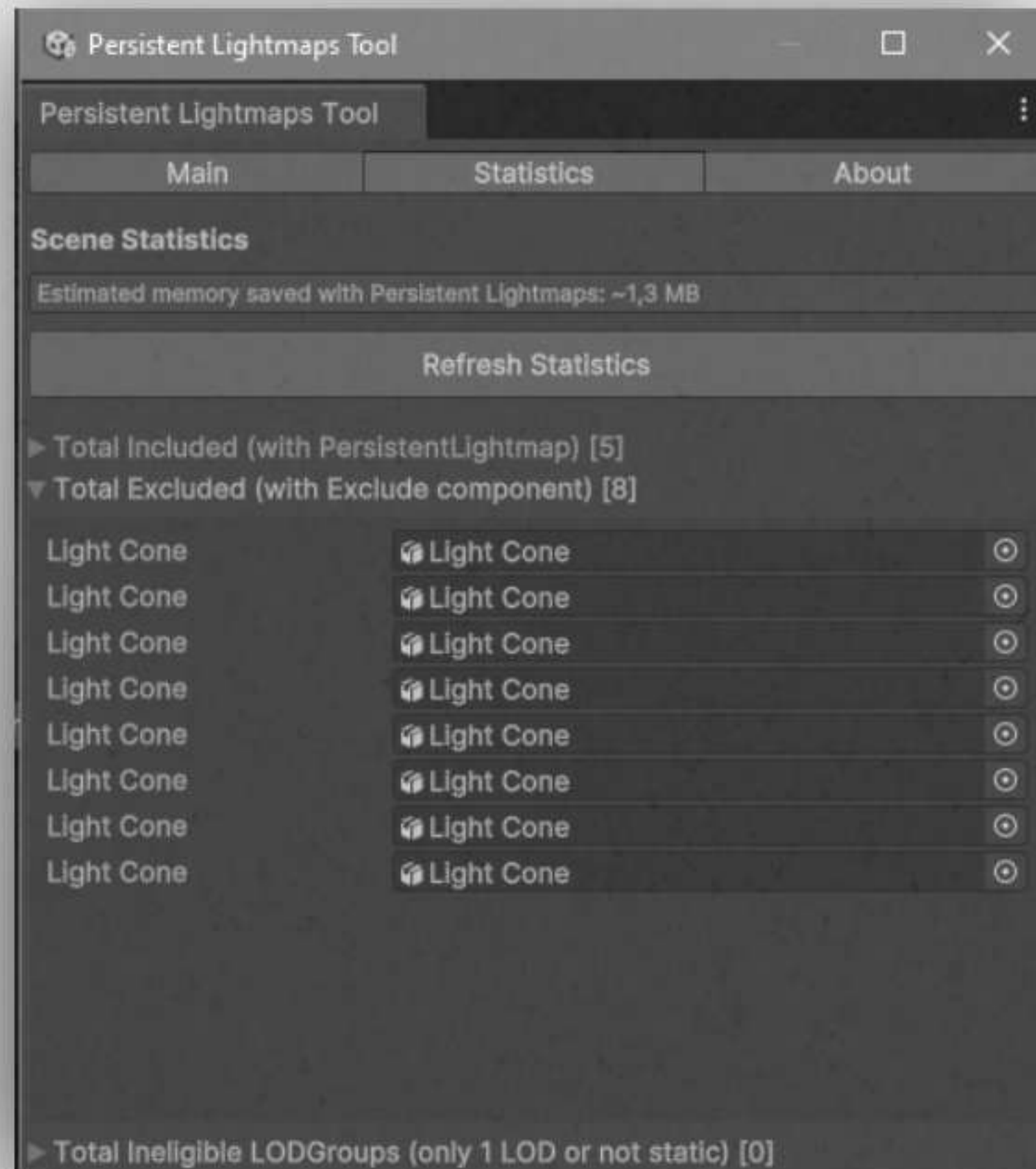
Excluding Objects

1.
Select the GameObject you want to exclude, make sure it has a LODGroup component.

2.
In the Inspector, click Add Component, then search for "ExcludeFromPersistentLightmap" and add it to the object.
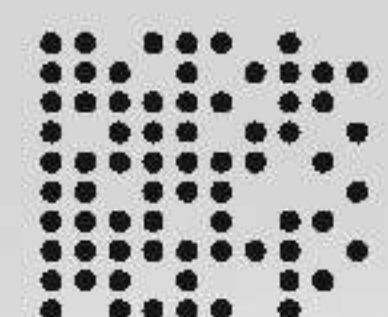
3.
Done! This object will now be ignored when you:

– Apply Persistent Lightmaps

– Remove Persistent Lightmaps

– Scan statistics (it will appear under "Total Excluded")

Note:
Removing the "ExcludeFromPersistentLightmap" component, will restore the LODGroup, and wont get skipped during processing.

Persistent Lightmaps Tool

| Main | Statistics | About |

Scene Statistics

Estimated memory saved with Persistent Lightmaps: ~1,3 MB

Refresh Statistics

▶ Total Included (with PersistentLightmap) [5]
▼ Total Excluded (with Exclude component) [8]

Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone
Light Cone	Light Cone

▶ Total Ineligible LODGroups (only 1 LOD or not static) [0]

The Persistent Lightmaps Tool

Developed by SWeeb

Usage 5:

Important Warning

This tool copies lightmap data from LOD0 to all other LODs assuming the different LOD meshes share identical or compatible lightmap UVs.
This is what makes the optimization possible, but it also comes with a critical limitation.

1.

If your LOD meshes have:
– Different lightmap UV unwraps
– Different UV island scales or positions
– Or missing lightmap UV channels

Then the copied lightmap data won't match the actual geometry.
This will cause:
– Visible lighting seams
– Incorrect shading
– Artifacts or light leaking, especially noticeable at lower LODs

2.

Lightmap data (index and scale/offset) only tells Unity which part of the baked lightmap to use. If the lower LOD mesh has different UVs, it will sample the wrong part of the texture, because the tool doesn't rebake or reunwrap anything, it simply reuses the LOD0 data.

Best Practice:

To use this tool correctly:
– Make sure all your LOD meshes share the same or very closely matched lightmap UVs.
– Check your imported FBX files to ensure LOD0, LOD1, LOD2, etc., have identical UV layouts for the lightmap channel.