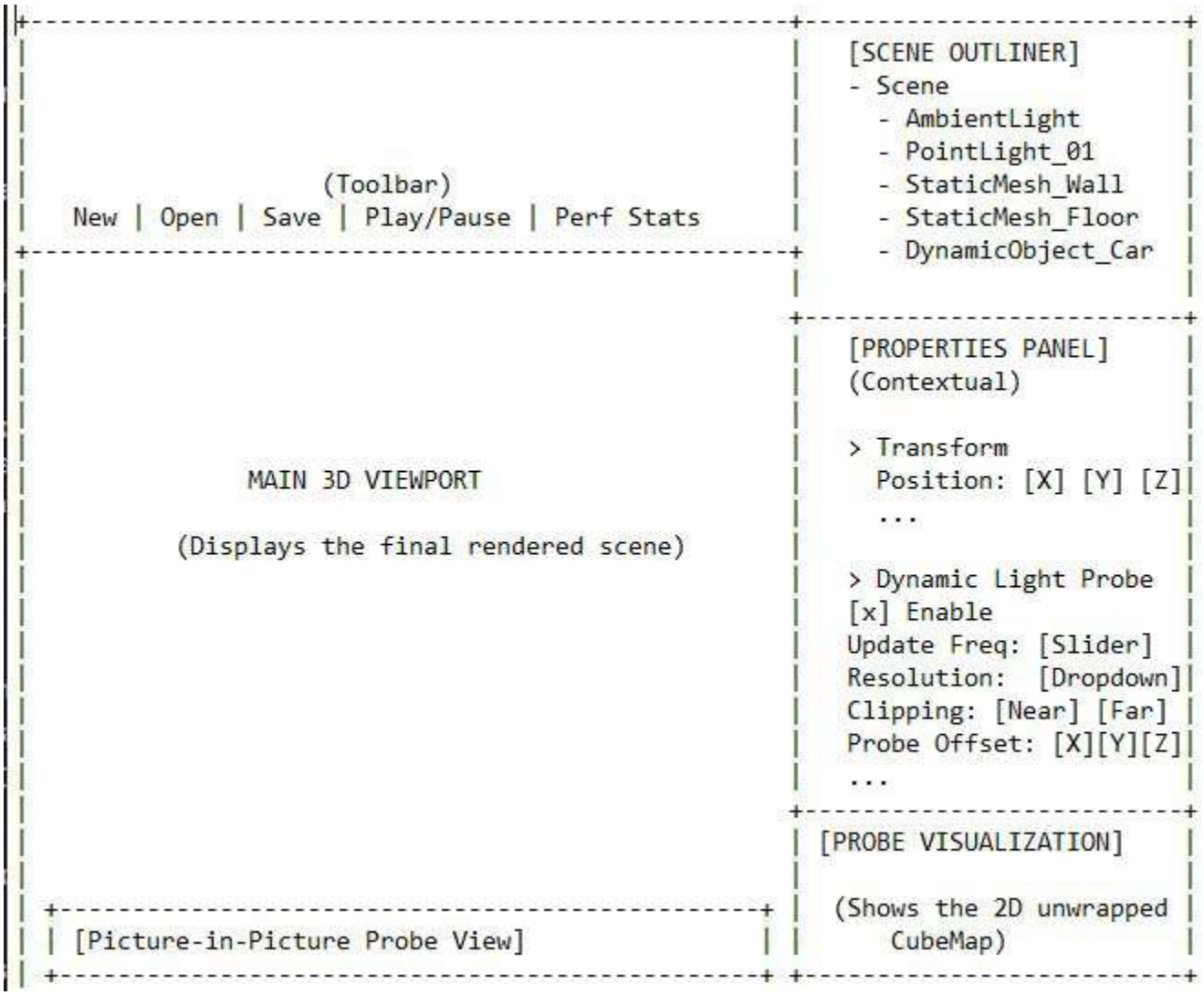
Light Probe.

1. The Core Concept

Your app's main job is to attach a CubeCamera to a moving object. On a controlled basis (e.g., every frame, every few frames, or when the object moves a certain distance), this CubeCamera will take a 360° snapshot of the scene from the object's perspective. This snapshot (a CubeMap) is then used to generate a LightProbe, which provides realistic, dynamic lighting and reflections for that object.

2. The App's UI/UX Mockup

Imagine a classic 3D editor layout. This provides a familiar and efficient workflow.



3. Detailed Breakdown of UI Panels

a) Main 3D Viewport
This is the primary view, showing the final result.
The user can orbit, pan, and zoom to inspect the scene.
When an object with a dynamic probe is moving, the user should see its reflections and lighting change in real-time according to its surroundings.
Picture-in-Picture (PiP) Probe View: A small overlay in a corner of the viewport that shows the live feed from the object's CubeCamera. This is essential for debugging and understanding what the probe is "seeing."
b) Scene Outliner
A standard tree view of all objects in the scene (lights, meshes, cameras).
Allows the user to easily select the object they want to apply the dynamic probe to (e.g., a moving car, a character).
c) Properties Panel (Context-Sensitive)
This is the heart of your app. When the user selects a mesh in the Outliner, this panel populates with its properties. You'll add a special section: "Dynamic Light Probe".
[x] Enable Dynamic Probe: The master switch. Toggling this on creates and attaches the CubeCamera and LightProbe to the selected object.
Update Frequency: Crucial for performance.
Options: "Every Frame," "Every 2 Frames," "On Move."
On Move Threshold: A distance value. The probe only updates if the object has moved more than this distance since the last update. This is a massive performance saver.
Resolution: The resolution of the CubeCamera's render target (the CubeMap).
Dropdown: 64x64, 128x128, 256x256, 512x512.
Why it's important: A 256x256 probe is rendered 6 times (once per face), which is expensive. Lower resolutions are much faster but provide blurrier reflections/lighting.
Clipping (Near/Far): The near and far planes for the probe's CubeCamera. Allows you to exclude objects that are too close or too far away from the probe's calculation.
Probe Offset: [X] [Y] [Z] input fields. Allows the user to offset the probe's capture position relative to the object's origin. Useful if the object's pivot point isn't in its visual center.
Intensity: A slider to control how much the probe affects the material's lighting.
Render Layer: An advanced but critical feature. A dropdown to select a specific render layer for the probe. This prevents the probe from "seeing" the object it's attached to, which would cause an ugly visual feedback loop.
d) Probe Visualization Panel
A dedicated, larger view that shows the currently generated CubeMap, often "unwrapped" into a cross or a lat-long format.
This helps diagnose issues like: "Why is that flare showing up in my reflection?" or "Is the clipping plane cutting off that building?"
4. The User Workflow
Load Scene: User opens a scene with static environment geometry and a dynamic object (e.g., a city block and a car).

Select Object: User clicks on the "Car" mesh in the Scene Outliner.
Enable Probe: In the Properties Panel, they check [x] Enable Dynamic Probe.
Observe: The car's material immediately looks better, reflecting the nearby buildings. The PiP view in the main viewport shows a 360° view from the car's position.
Animate/Move: The user presses "Play" or manually drags the car. As the car moves down the street, its reflections change in real-time.
Optimize: The user notices a frame rate drop. They go to the Properties Panel and change the Update Frequency from "Every Frame" to "On Move" with a threshold of 0.5 units. The performance instantly improves, and the visual quality is still excellent because the lighting only updates when needed.
Fine-Tune: The user notices the car's own hood is appearing in the reflection. They create a new render layer for everything except the car, and then assign that layer to the probe's Render Layer property. The artifact disappears.