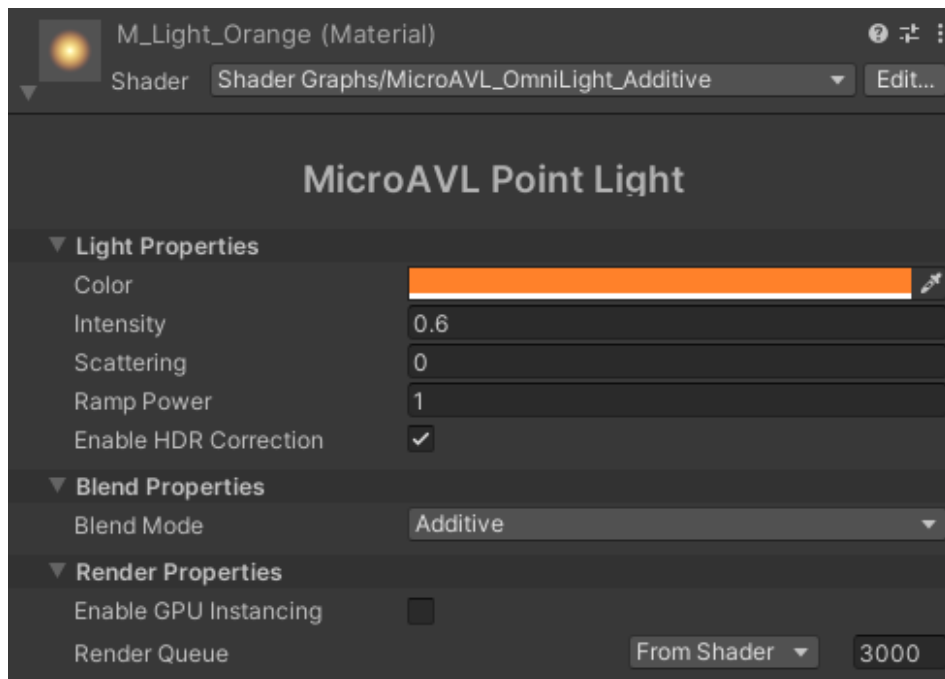
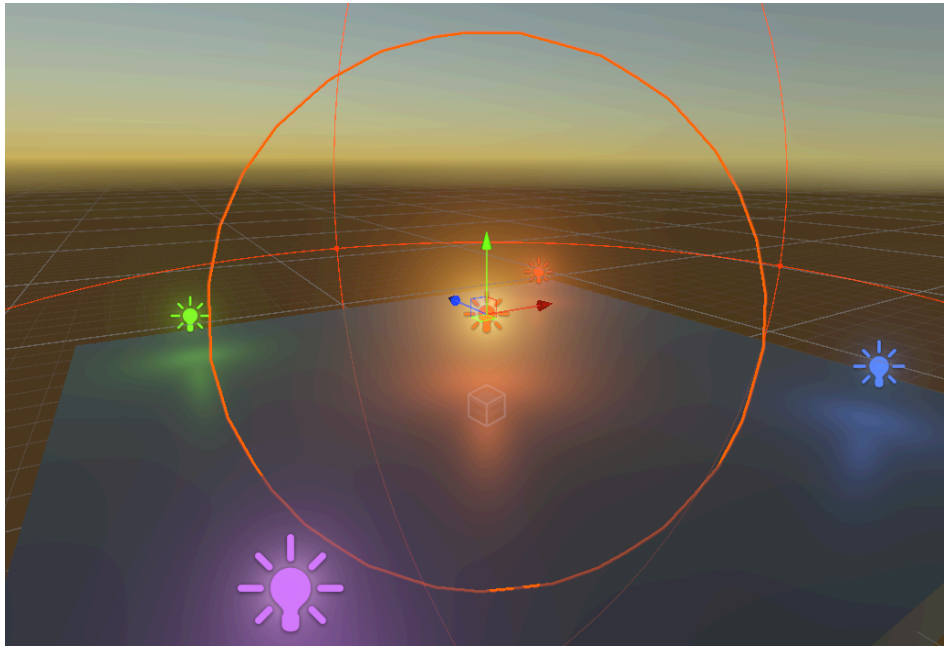


MicroAVL v1.0

1. Setup

Setup of the shader is straightforward: just place a sphere with the **AVL shader** assigned to the material. As a shortcut, you can use the menu button: **GameObject / Light / MicroAVL Light**

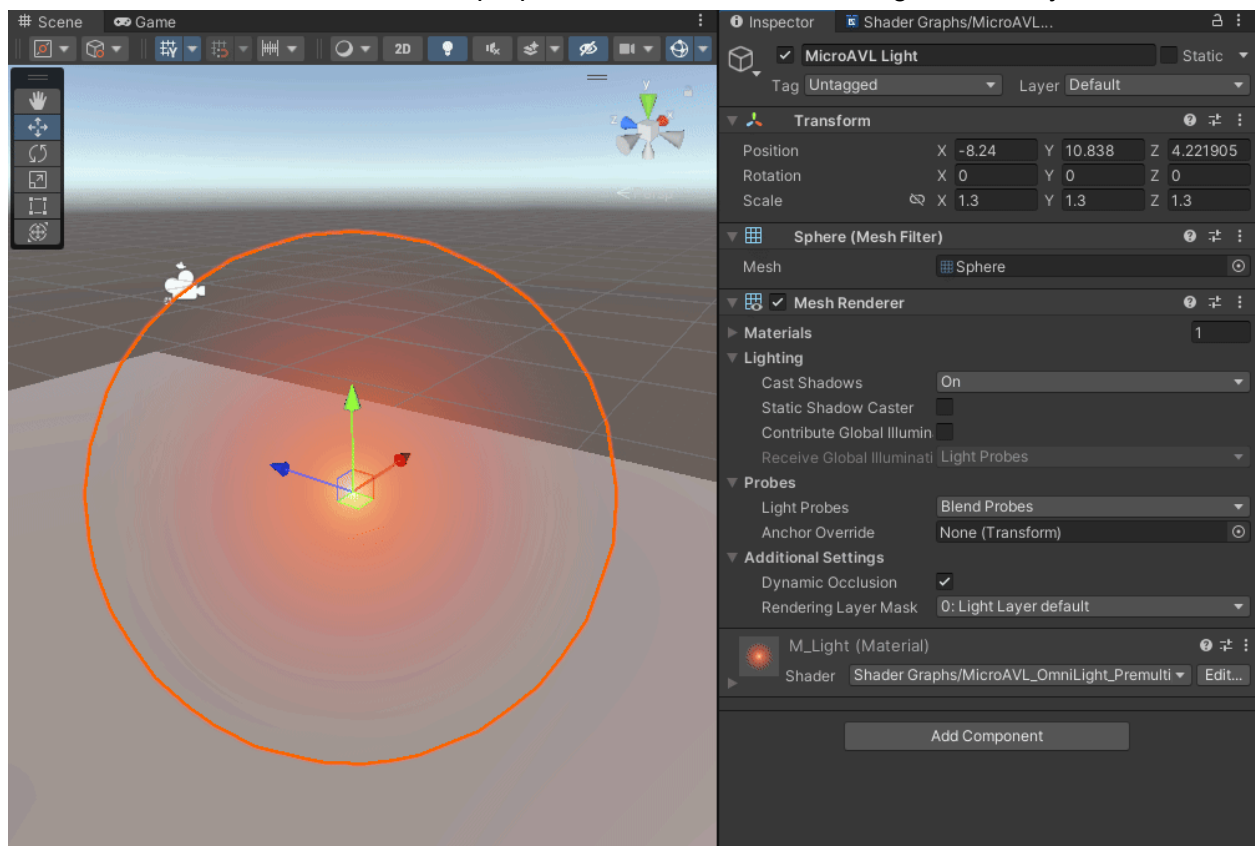


Shader Properties

- **Color:** Color of the light
- **Intensity:** Intensity of the light
- **Scattering:** 'Scattering' or 'blurring' factor of the light (makes light more evenly distributed in space)
- **Ramp Power:** a pow() function factor used to fine-tune the final look of the light
- **Enable HDR Correction:** Divides the final light calculation result by itself plus one. Makes the light look much more natural
- **Blend Mode:** changes what blend mode is being applied

2. Tips

- a. You can use the constrained proportions button to scale the light efficiently:



- b. Additive blend mode is mostly the way to go. Even though premultiply blend might look better in some cases, it causes rendering conflicts when there is an intersecting light sphere geometry.
- c. Due to some Unity limitations, light shader is rendered even behind walls. This means that even if light is not visible, it might still be rendered and take some performance. If

that's a serious problem for your project, take a look at our full [Analytical Volumetric Lighting](#) solution which addresses this problem.

3. F.A.Q.

Q: What is AVL, how does it work?

A: AVL stands for Analytical Volumetric Lighting. Basically, it's an analytical solution for the volumetric lighting equation, as opposed to the raymarching solutions where everything is approximated. It requires less computational resources, doesn't introduce noise, but also doesn't produce any shadows.

Q: Is it possible to cast volumetric shadows?

A: No, it won't work with the analytical approach

Q: Something is not working and I cannot understand why

A: You can contact me via my email: akidevcat@gmail.com

Q: Is it possible to use this shader in my commercial project?

A: Yes

Q: What platforms are supported?

A: Basically, everything that runs URP and its shaders, except for some VR