Full Rendering Equation

In our project we used `THREE.MeshStandardMaterial`, looking at `bsdfs.glsl.js` and `envmap_fragment.glsl.js` the rendering equation used for this material is:

$$L(l, geometry, material) =$$

$$\pi * (irradiance * BRDF_{specularGGX}(roughness, c_{spec}, l, v, n) + irradiance * BRDF_{lambert}(c_{diff}))$$

$$+$$

$$envColor * specularStrength * reflectivity$$

BRDF lambert

This is the diffuse component of the BRDF:

$$BRDF_{lambert}(c_{diff}) = c_{diff}/\pi$$

BRDF specular GGX

This is the specular component of the BRDF:

$$BRDF_{specularGGX} = F_{Schlick}(c_{spec}, l \cdot h) * G_{GGX}(\alpha, n \cdot l, n \cdot v) * D_{GGX}(\alpha, n \cdot h)$$

$$h = l \cdot v$$

Fresnel term (Schhlick approximation)

$$F_{Schlick}(c_{spec}, l \cdot h) = (1 - c_{spec})(2^{(l \cdot h)(-5.55(l \cdot h) - 6.98)}) + c_{spec}$$

Geometry function (Smith)

$$G_{GGX}(\alpha, n \cdot l, n \cdot v) = \frac{1}{2 \cdot max(gv + gl, \varepsilon)}$$

$$gv = (n \cdot l)\sqrt{2^{\alpha} + (1 - 2^{\alpha}) * 2^{n \cdot v}}$$

$$gl = (n \cdot v)\sqrt{2^{\alpha} + (1 - 2^{\alpha}) * 2^{n \cdot l}}$$

$$\alpha = roughness^2$$

Normal distribution function

$$D_{GGX}(\alpha, n \cdot h) = \frac{1}{\pi} * \frac{2^{\alpha}}{2^{2^{(n \cdot h)}(2^{\alpha}-1)+1}}$$

Environment color

La componente $envColor$ si ottiene accedendo alla cube environment map tramite il view vettore riflesso rispetto la normale, come visto a lezione e contenuto nel file `envmap_fragment.glsl.js`.