

## **AA1-Shaders Documentació**

### **Blur**

Apliquem “Gaussian Blur”, aquest mètode aplica píxels amb un desplaçament més petit al centre una prioritat més baixa, on es calcula el pes de cada píxel amb una funció de gauss

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Primer calculem el quadrat de la **desviació estàndard** i després calculem la funció en si, la primera meitat de l'esquerra, dividim 1 per l'arrel quadrada de dues vegades pi pel quadrat de la desviació estàndard. Després ho multipliquem amb la part dreta que és el nombre d'Euler per la potència del menys desplaçament al quadrat dividit per 2 vegades la desviació estàndard al quadrat.

L'efecte desitjat acaba donant una vista borrosa, aquest tipus d'efecte pot servir per a simular atordiments en jugadors.

### **Vignette**

El Vignette comença agafant el color original del píxel.

Un cop fet això, mapegem la coordenada respecte a la posició cèntrica de la pantalla, afegint una variable “\_center” que permetrà canviar el punt cèntric de l'efecte des de l'inspector.

La següent línia ens serveix per a fer que el mapeig anterior es torni més o menys quadrat elevat la coordenada a la variable “\_roundness”. Seguidament, calcularem el factor, que ens servirà per a saber quant afectat es veurà el píxel. Aquí és on modifiquen la intensitat i com es veurà afectat cada axis amb les seves respectives variables.

Després apliquem el color dels píxels i fem un lerp en funció de “\_blend”, i per acabar canviem el color pel qual vulguem a l'exterior i tornem el color final.

## BRDF

Per a fer Fresnel hem utilitzat “Fresnel Schlick” en comptes del tradicional per tal d'optimitzar el procés.

$$F_{Schlick}(q, l, h) = q + ((1 - q)(1 - h \cdot l))^5$$

Per a la component geomètrica hem escollit la fórmula de la Geometria implícita, ja que era la que més barata semblava en l'àmbit computacional.

$$G_{Implicit}(l, h, v) = (n \cdot l)(n \cdot v)$$

En el cas de la funció de distribució, ens hem decantat per la funció GGX, perquè la llegibilitat ens va semblar millor.

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

Seguim fent servir la fórmula de BRDF i aplicant el resultat al color final.

$$f(\mathbf{l}, \mathbf{v}) = \frac{F(\mathbf{l}, \mathbf{h})G(\mathbf{l}, \mathbf{v}, \mathbf{h})D(\mathbf{h})}{4(\mathbf{n} \cdot \mathbf{l})(\mathbf{n} \cdot \mathbf{v})}$$

## Fórmula global BRDF:

$$\begin{array}{c}
 \text{paràmetre} + (1 - \text{paràmetre}) * (1 - \text{dot}(\text{punt de llum}, \text{half vector}))^5 \quad \alpha^2 / \pi * (\text{dot}(\text{normal}, \text{half})^2 * (\alpha^2 - 1) + 1)^2 \\
 \downarrow \qquad \qquad \qquad \downarrow \qquad \qquad \qquad \downarrow \\
 \text{Fresnel}(\text{vector de llum}, \text{half vector}) * \text{Geometry}(\text{Punt de llum}, \text{vector vista}, \text{half vector}) * \text{Funció de distribució}(\text{half vector}) \\
 \hline
 4 * \overset{(\text{dot})}{(\text{normal} * \text{vector llum})} * \overset{(\text{dot})}{(\text{normal} * \text{vector vista})}
 \end{array}$$