
Bloom

Sugar Framework



Figure 0



Figure 1



Figure 2



Figure 3



Figure 4



Figure 5

La réalisation de l'effet nécessite de créer plusieurs textures. Une texture principale RT pour le rendu des objets de la scène et un ensemble de mipmaps M_0 à M_i nécessaire à la dilatation de la lumière et au flou.

Chaque objet est rendu dans RT avec sa lightmap L et sa diffuse D (figure 0).

La couleur rgb résultante est un simple multitexturing multiplicatif entre D et L .

Nous stockons dans le canal alpha de RT l'intensité de la lightmap i_L au-delà d'un seuil th . Le seuil peut être variable selon le résultat final souhaité.

$$i_L = \max(L_r, L_g, L_b) - th$$

Le but de cette opération est d'obtenir dans M_0 une information mélangeant la couleur finale d'un pixel et son intensité lumineuse ci (figure 1)

$$ci = D * L * i_L$$

Le bloom est donc directement dépendant de l'intensité de la lightmap afin de laisser à l'artiste tout control possible lors de la création.

On applique ensuite une dilatation de la lumière sur ci (figure 2)

C'est sur cette dilatation qu'un flou gaussien est effectué (figure 3) .

Dans Sugar, le flou gaussien est effectué en mélangeant une image avec ses propres mipmap, ce qui permet d'appliquer des gaussiennes avec un large rayon sans avoir un nombre de passes d'affichage élevé.

La dernière étape est finalement un mélange entre l'image de base et notre image floutée ainsi obtenue. La formule est de la forme

$$C_0 * f_0 + C_1 * f_1$$

La couleur C_0 correspond à l'image de base

La couleur C_1 correspond à notre image floutée en tenant compte essentiellement des halos lumineux. Pour ce faire l'intensité de chaque pixel est inversement proportionnelle à son intensité de base i_L (figure 4)

Les facteurs f_0 et f_1 permettent de garder un control sur le mélange des 2 textures, pour avoir plus ou moins de bloom, plus ou moins d'intensité lumineuse

Le résultat obtenu (figure 5) est donc finalement rendu par :

$$D * L * f_0 + Blur(Dil(D * L * i_L)) * (1 - i_L) * f_1$$