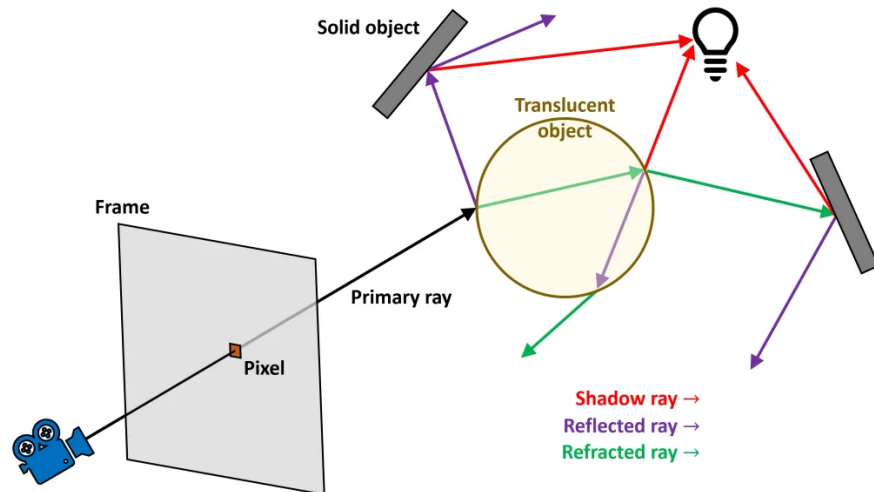


Algorithme Ray Tracer



<https://www.techspot.com/article/1888-how-to-3d-rendering-rasterization-ray-tracing/>

Nous utilisons un algorithme principal du lanceur de rayon. Il s'agit de voir le monde avec les yeux plutôt que de voir comment le monde parvient aux yeux. Du côté de la vue, nous envoyons un rayon lumineux vers chaque pixel de l'image, nous traçons ce chemin lumineux et calculons la direction de son rayon inverse, en le mettant sur le pixel correspondant. En calculant la décroissance de la couleur et la superposition sur le trajet de la lumière, la couleur de chaque pixel peut être essentiellement déterminée.

Le traçage de rayons peut être considéré comme un processus récursif. Un rayon est émis dans la scène, l'intersection la plus proche entre le rayon et la géométrie est trouvée, et si le matériau à cette intersection est réfléchissant ou réfringent, le rayon peut être tracé dans la direction de la réflexion ou de la réfraction, et ainsi de suite, de manière récursive, jusqu'à la source de lumière (ou la couleur de fond), et la valeur de coloration de chaque pixel est calculée.

Algorithme Récursif Ray Tracer :

Input: rayon lumineux

Output: couleur du rayon inverse

Function tracing():

Si non intersection avec aucun objet alors
retourner couleur de fond

Sinon

obj ← trouver l'objet le plus proche du rayon;

rayon réfléchissant ← getRayonReflect(obj);

rayon réfractaire ← getRayonRefract(obj);

couleur principal ← le rayonnement de l'obj;

couleur réfléchissant ← tracing(rayon réfléchissant);

couleur réfractaire ← tracing(rayon réfractaire);

retourner mix(couleur principal, couleur réfléchissant, couleur réfractaire);