

# Explicación Detallada de Funciones para Raytracing

## 1. Funciones de Vectores

- `vector_add`: Suma dos vectores. Usado para calcular posiciones y trayectorias.
- `vector_sub`: Resta dos vectores. Usado para calcular dirección entre puntos.
- `vector_scale`: Escala un vector por un escalar. Usado para mover posiciones con rayos.
- `vector_dot`: Producto escalar. Usado para ángulos, normalización y ecuaciones.
- `vector_normalize`: Normaliza un vector. Necesario para tener direcciones unitarias.
- `vector_cross`: Producto vectorial. Usado para construir el sistema de coordenadas de cámara.

## 2. Cámara y Proyección

- `get_ray_direction`:

Calcula la dirección de un rayo para un píxel dado.

Construye una base ortonormal (right, up, forward) y aplica proyección en perspectiva.

## 3. Intersección

- `intersect_sphere`:

Calcula si un rayo intersecta con una esfera.

Resuelve una ecuación cuadrática con la forma:  $\|O + tD - C\|^2 = R^2$ .

## 4. Renderizado

- `set_pixel`:

Dibuja un píxel en el buffer de imagen con el color calculado.

Usado al final del proceso si se determina que un rayo golpea un objeto.

## 5. Conexión General

- Por cada píxel (x, y):
  - `get_ray_direction()` calcula la dirección del rayo desde cámara.
  - `intersect_sphere()` verifica si golpea una esfera.
  - Si hay intersección, se llama a `set_pixel()` con el color del objeto.

Este flujo permite simular un entorno 3D básico usando raytracing.