

MiniRT - Guía Paso a Paso

Introducción y Estructura

MiniRT es un proyecto de la escuela 42 donde se implementa un motor de ray tracing básico en C. Su propósito es renderizar escenas 3D descritas en archivos `.rt``, utilizando operaciones matemáticas con vectores, intersecciones y modelos de iluminación.

Pasos iniciales:

1. Lee el subject completo.
2. Crea tu estructura de carpetas: `includes/`, `src/`, `parser/`, `math/`, `render/`.
3. Define tus estructuras (`scene`, `color`, `vector`, `cámara`, etc.).
4. Implementa un parser para leer los archivos `.rt``.
5. Implementa funciones para cada tipo de entidad: esfera, plano, cilindro, luz, cámara.
6. Usa MiniLibX para mostrar la imagen renderizada.

Álgebra, Intersecciones e Iluminación

1. Vectores 3D:

- Operaciones: suma, resta, producto escalar (dot), producto vectorial (cross), normalización, etc.

2. Rayos:

- Un rayo se define por: $\text{origen} + t * \text{dirección}$.

3. Intersecciones:

- Esfera: resuelve ecuación cuadrática para hallar t .
- Plano: usa dot product para obtener t .
- Cilindro: se necesita más álgebra, considera casos infinitos y recorte.

4. Iluminación (modelo Phong):

- Ambient: luz base aplicada a toda la escena.
- Diffuse: depende del ángulo entre la luz y la superficie.

MiniRT - Guía Paso a Paso

- Specular: brillo simulado con potencia del coseno.

5. Sombras:

- Lanza rayos hacia las luces. Si otro objeto bloquea, solo aplica luz ambiente.

6. Colores:

- Usa componentes RGB y combina resultados de iluminación.

Funciones clave y Tips

Funciones a implementar:

- Vector: `vec_add`, `vec_sub`, `vec_dot`, `vec_cross`, `vec_normalize`.
- Rayos: estructura Ray con origen y dirección.
- Intersecciones: intersección con esfera, plano, cilindro.
- Iluminación: calcular color final con Phong.
- Parser: interpretar líneas como "A 0.2 255,255,255".

Tips:

- Usa `printf` para depurar coordenadas e intersecciones.
- Implementa función para liberar arrays dinámicos (`free_split`).