

. Introdução

Esta aplicação é um exemplo de **ray casting**, um método gráfico utilizado para simular a projeção de raios em um ambiente 3D e gerar imagens 2D a partir dessas projeções. A aplicação é implementada utilizando duas linguagens de programação:

- **C++**: para a lógica de ray casting e geração da imagem.
- **Python**: para a visualização da animação.

2. Estrutura da Aplicação

2.1. C++: Lógica de Ray Casting

O código em C++ é responsável por realizar o cálculo de ray casting e gerar a imagem com base na interseção dos raios com esferas no espaço 3D. As principais funções e métodos utilizados são:

- Função **subtract(Vec3 a, Vec3 b)**: Subtrai dois vetores 3D.
- Função **dot(Vec3 a, Vec3 b)**: Calcula o produto escalar entre dois vetores 3D.
- Função **intersect_ray_sphere(Vec3 origin, Vec3 direction, Sphere sphere, double *t)**: Determina se um raio intercepta uma esfera e calcula o valor **t** da interseção.
- Função **generate_ray_casting(int width, int height, Sphere* sphere, int* output)**: Gera a imagem 2D do ray casting, preenchendo um array com valores de pixel.

Código C++:

- **raycasting.h**: Define as estruturas e declara as funções.
- **raycasting.cpp**: Implementa a lógica das funções de ray casting.

2.2. Python: Visualização e Animação

O código em Python utiliza a biblioteca **matplotlib** para criar uma animação que exibe a esfera se movendo ao longo do eixo z. As principais bibliotecas e métodos utilizados são:

- **ctypes**: Carrega a biblioteca compartilhada (**libraycasting.so**) e define as estruturas e funções importadas.
- **numpy**: Manipula os dados da imagem.
- **matplotlib**: Cria e exibe a animação.

Código Python:

- Função `generate_image(width, height, sphere)`: Chama a função C++ para gerar uma imagem com base na esfera e seus parâmetros.

4. Conclusão

Esta aplicação demonstra como integrar C++ e Python para realizar cálculos gráficos e gerar animações. A lógica complexa de ray casting é manipulada eficientemente em C++, enquanto Python é utilizado para a visualização e animação, aproveitando suas bibliotecas de alto nível.