

# Ray Tracing: O Mundo Através De Raios de Luz

XXXVI Jornada Giulio Massarani de Iniciação Científica, Tecnológica,  
Artística e Cultural

**Thiago Barroso Perrotta**  
Prof.<sup>o</sup> Ricardo G. marroquim

Universidade Federal do Rio de Janeiro

10 de outubro de 2014



# Agenda

- 1 Raytracer
- 2 Extração de primitivas em nuvens de pontos
- 3 Resultados
- 4 Referências

# Conceituando

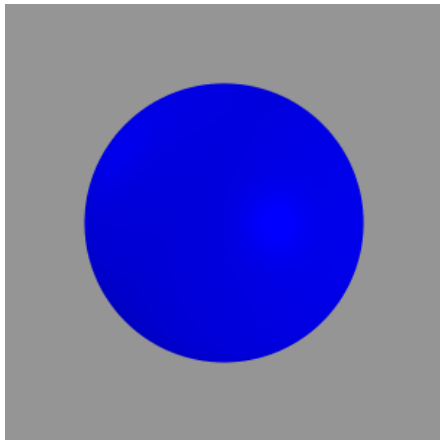
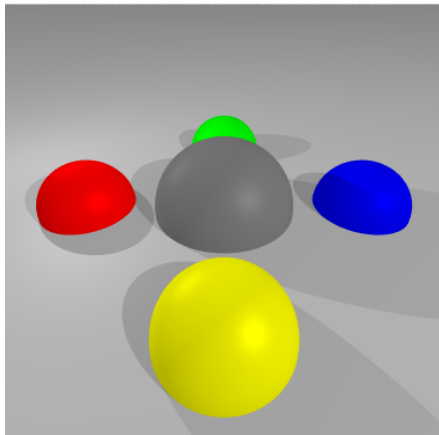
- Renderizar imagens

# Algoritmo

- Defina alguns objetos
- Especifique um material para cada objeto
- Defina algumas fontes de luz
- Defina uma janela cuja superfície seja coberta com pixels
- Para cada *pixel*
  - atire um raio, a partir do centro do *pixel*, na direção dos objetos
  - compute, dentre os pontos atingidos nos objetos, o que esteja mais próximo
  - Se o raio atingiu um objeto
    - use o material do objeto e as fontes de luz para computar a cor do pixel
  - Senão
    - defina a cor do *pixel* como preta

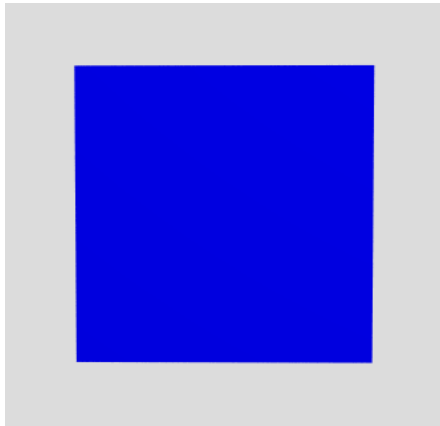
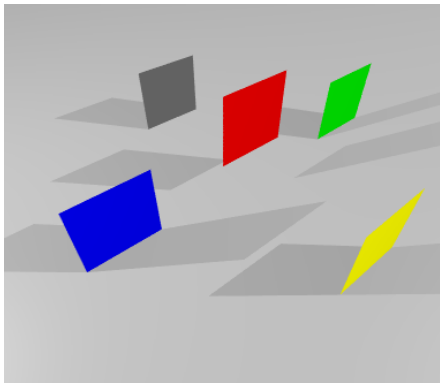
# Objetos

## Esferas



# Objetos

## Retângulos



# Objetos

## Triângulos

# Objetos

Toros



# Algoritmo

- Defina alguns objetos
- Especifique um material para cada objeto
- Defina algumas fontes de luz
- Defina uma janela cuja superfície seja coberta com pixels
- Para cada *pixel*
  - atire um raio, a partir do centro do *pixel*, na direção dos objetos
  - compute, dentre os pontos atingidos nos objetos, o que esteja mais próximo
  - Se o raio atingiu um objeto
    - use o material do objeto e as fontes de luz para computar a cor do pixel
  - Senão
    - defina a cor do *pixel* como preta

# Materials

Materials

# Algoritmo

- Defina alguns objetos
- Especifique um material para cada objeto
- Defina algumas fontes de luz
- Defina uma janela cuja superfície seja coberta com pixels
- Para cada *pixel*
  - atire um raio, a partir do centro do *pixel*, na direção dos objetos
  - compute, dentre os pontos atingidos nos objetos, o que esteja mais próximo
  - Se o raio atingiu um objeto
    - use o material do objeto e as fontes de luz para computar a cor do pixel
  - Senão
    - defina a cor do *pixel* como preta

# Fontes de luz

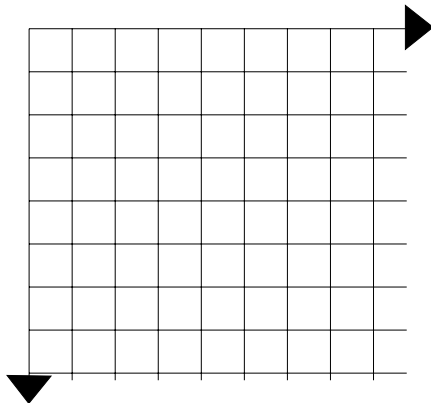
Fontes de luz

# Algoritmo

- Defina alguns objetos
- Especifique um material para cada objeto
- Defina algumas fontes de luz
- Defina uma janela cuja superfície seja coberta com pixels
- Para cada *pixel*
  - atire um raio, a partir do centro do *pixel*, na direção dos objetos
  - compute, dentre os pontos atingidos nos objetos, o que esteja mais próximo
  - Se o raio atingiu um objeto
    - use o material do objeto e as fontes de luz para computar a cor do pixel
  - Senão
    - defina a cor do *pixel* como preta

# Plano de visualização

- Número de *pixels* (ex.: 400x400)
  - Horizontal
  - Vertical
- Tamanho de cada *pixel*  $\implies$  *zoom*



# Algoritmo

- Defina alguns objetos
- Especifique um material para cada objeto
- Defina algumas fontes de luz
- Defina uma janela cuja superfície seja coberta com pixels
- Para cada *pixel*
  - atire um raio, a partir do centro do *pixel*, na direção dos objetos
  - compute, dentre os pontos atingidos nos objetos, o que esteja mais próximo
  - Se o raio atingiu um objeto
    - use o material do objeto e as fontes de luz para computar a cor do pixel
  - Senão
    - defina a cor do *pixel* como preta

# Interseção entre raio e objetos

- Função *Hit* para cada objeto
-





# Resultados

# Conclusão

- Raytracer
- Detecção de formas em nuvens de pontos

# Trabalho Futuro

# Referências

-  Suffern, Kevin Geoffrey, and Suffern, Kevin. Ray Tracing from the Ground up. AK Peters, 2007.
-  Schnabel, Ruwen, Roland Wahl, and Reinhard Klein. "Efficient RANSAC for Point-Cloud Shape Detection." Computer graphics forum. Vol. 26. No. 2. Blackwell Publishing Ltd, 2007.