

---

## Introdução à Computação Gráfica

Emmanuella Faustino Albuquerque  
20170002239

### Atividade Prática 6: Modelagem e Renderização em Ray Tracing 29 de novembro de 2021

#### VISÃO GERAL

Nesta atividade foram construídas algumas cenas utilizando o sistema de rendering baseado em ray tracing: o POV-Ray.

#### ESTRATÉGIAS

##### Familiarizando-se com o Pov-Ray

Com base na documentação do pov-ray ([Getting Started](#)), temos que para criar as cenas precisamos somente utilizar a linguagem de script própria do render utilizando um arquivo com a extensão (.pov).

No caso de sistemas Linux é possível executar utilizando o comando abaixo:

---

```
// Executa o arquivo (.pov) gerando a imagem final.  
povray Bunny.pov
```

---

#### RESULTADOS

##### Cena 1: Baseball Bat

Para construir essa cena foi preciso primeiramente definir algumas esferas. Sendo 5 no total, onde 4 são utilizadas para construir o taco de beisebol e a última para a bola.

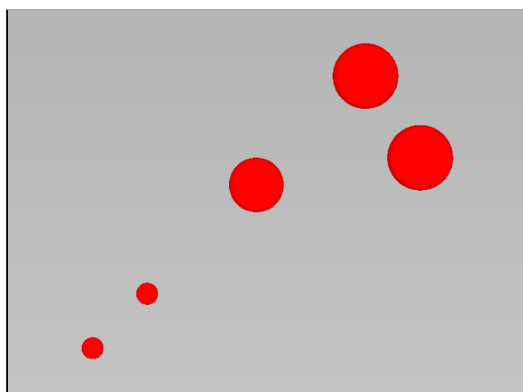


Imagem 1: Definindo os pontos(esferas)

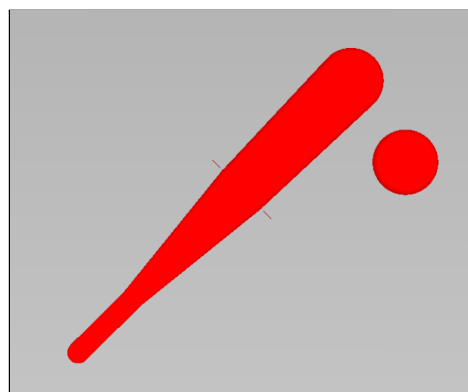


Imagem 2: Um taco de beisebol e sua bola

Definido os pontos, isto é, as esferas, foi possível utilizar os chamados “Spline Identifiers” do pov-ray, que oferecem uma maneira de definir caminhos(pathways) nas cenas. Com isso, utilizando o “Sphere Sweep Object” foi possível conectar as esferas criando o taco de beisebol.

---

```
sphere_sweep {  
  linear_spline // linear ( straight ) path  
  4, // positions  
  
  // ( position , radius )  
  <-2, -2, 0>, .2 // sphere 1: start  
  <-1, -1, 0>, .2 // sphere 2  
  <1, 1, 0>, .5 // sphere 3  
  <3, 3, 0>, .6 // sphere 4: end  
  
  ...}
```

---



Imagem 3: Um taco de beisebol e uma bola com textura

Inicialmente foi utilizado pigmento vermelho nas esferas, mas para finalizar foi adicionada uma textura de madeira para o taco e uma outra textura para a bola.

## Cena 2: Bunny

Primeiramente foi preciso obter a representação da malha do coelho de stanford, felizmente, o Pov-Ray utiliza o “Mesh2 Object” e já existe uma representação da malha do coelho em mesh2 onde todos os vértices são listados utilizando vertex\_vectors.

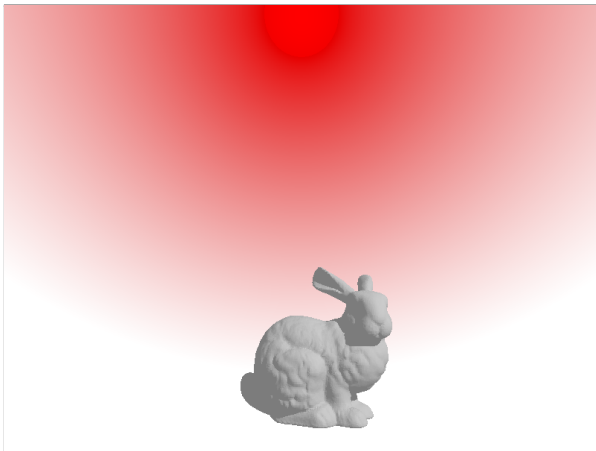


Imagem 4: Um coelho com um céu sky\_sphere

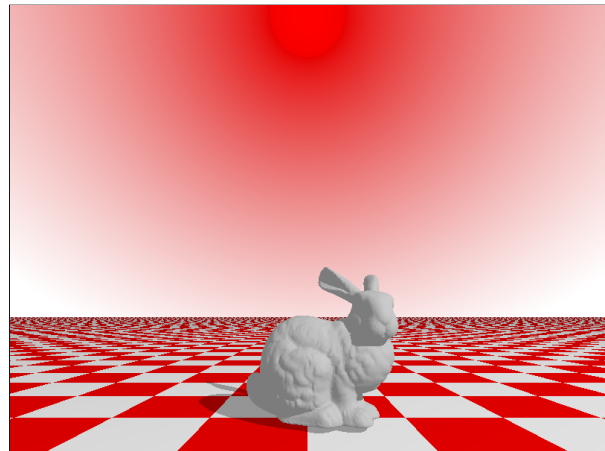


Imagem 5: Um coelho em um plano com padrão xadrez

Com o coelho posicionado, foi então definido o plano, e utilizando os “Pattern Modifiers: turbulence, frequency, entre outros” foi possível chegar na imagem abaixo.

---

```

plane {
  y, 0
  pigment {
    checker
    pigment { Red }
    pigment { White }
    turbulence 0.5
    lambda 1.5
    omega 0.8
    octaves 5
    frequency 3
  }
  ...}

```

---

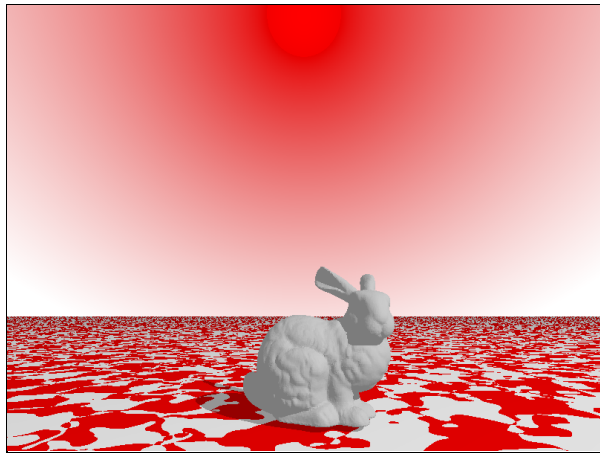


Imagem 5: Um coelho e um céu vermelho

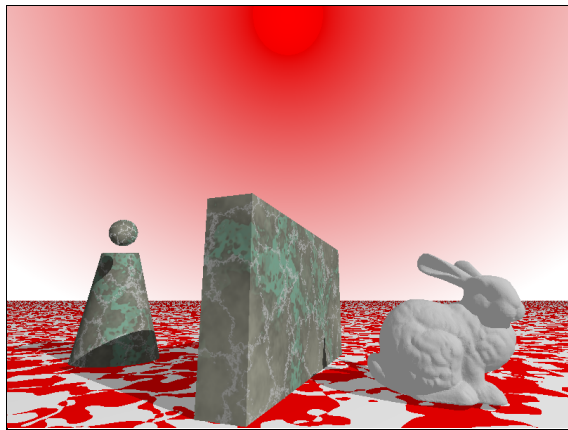


Imagem 5: Um coelho com alguns formas básicas

Por fim, com o intuito de testar algumas formas básicas disponíveis no Pov Ray, foram adicionados um cone, uma esfera e um box object, utilizando uma textura pré definida.

---

```

texture {
  T.Stone25    // Pre-defined from stones.inc
  ...}

```

---

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Spline Identifiers. POV-Ray 3.6 Documentation Online View. Disponível em: <https://www.povray.org/documentation/view/3.7.0/234/>. Acesso em: 24 de novembro de 2021.
- [2] Sphere Sweep Object. POV-Ray 3.6 Documentation Online View. Disponível em: <https://www.povray.org/documentation/view/3.7.1/63/>. Acesso em: 24 de novembro de 2021.
- [3] Bitmap Modifiers. POV-Ray 3.6 Documentation Online View. Disponível em: <https://www.povray.org/documentation/view/3.7.0/408/>. Acesso em: 24 de novembro de 2021.
- [4] Stanford Bunny. POV-Ray 3.6 Documentation Online View. Disponível em: <https://news.povray.org/povray.binaries.misc/message/%3C5432afe4@news.povray.org%3E/>. Acesso em: 24 de novembro de 2021.