

MC937A/MO603A – Computação Gráfica - 2020-S2 - Jorge Stolfi

Trabalho de laboratório 01 - 2020-10-07

Virus Mutante

Objetivos. treinar uso dos principais objetos geométricos primitivos do POV-Ray e desenvolver intuição sobre coordenadas cartesianas no espaço.

Enunciado. Todo mundo já cansou de ver ilustrações do vírus da COVID-19. Mas vírus facilmente sofrem mutações; quanto mais pessoas esse vírus infectar, maior será a probabilidade de aparecer um mutante que ninguém reconheceria se encontrasse na rua, no elevador, ou numa festinha de aniversário.

Sua tarefa nesta aula prática é produzir uma ilustração POV-Ray de um possível mutante do vírus da COVID-19, para que as autoridades sanitárias não percam tempo examinando vídeos de segurança e produzindo retratos falados, se e quando ele surgir.

Parte 1. Antes de começar a programar, **desenhe um esboço do seu vírus mutante e apresente-o ao professor via Meet, quando solicitado, no início da aula.**

Parte 2. Produza uma imagem da sua cena usando POV-ray. O modelo deve usar **pelo menos sete sólidos geométricos primitivos** do POV-Ray dentre sphere, cylinder, cone e box, combinados com operações booleanas. O modelo deve usar **pelo menos um sólido de cada tipo.**

Arquivos. Copie os arquivos da aula passada para uma nova sub-pasta 2020-10-07 da pasta mc937 no seu computador. Edite o arquivo main.pov, trocando a cena da aula anterior pela descrição de seu vírus mutante. Execute o comando **make** numa shell para gerar a imagem.

Originalidade. O arquivo de descrição main.pov deve ser construído manualmente, com um editor de texto comum, **sem** o auxílio de qualquer editor gráfico ou outra ferramenta de modelagem geométrica. Não é permitido copiar ou incluir quaisquer arquivos POV-Ray além dos fornecidos pelo professor ou escritos por você mesmo. Porém, é permitido re-usar arquivos ou trechos de código de exercícios anteriores.

Individualidade. Lembre-se de que todo trabalho prático é **individual**. Não é permitido pedir qualquer tipo de ajuda a colegas ou outras pessoas. Dúvidas devem ser tiradas apenas com o professor.

Exportação. Não se esqueça de **exportar seu arquivo main.pov até o final da aula para sua pasta WWW pública**

http://students.ic.unicamp.br/~raSEU_RA/mc937-2020-2/2020-10-07/

Comandos. Os comandos POV-Ray que produzem os sólidos necessários são:

- `sphere{ < Xc, Yc, Zc >, R texture{ Tx } }`
Este comando acrescenta à cena uma esfera. O centro é o ponto de coordenadas cartesianas (Xc, Yc, Zc) , e o raio é R .
- `cylinder{ < Xa, Ya, Za >, < Xb, Yb, Zb >, R texture{ Tx } }`
Este comando acrescenta um cilindro em posição arbitrária. O ponto de coordenadas cartesianas (Xa, Ya, Za) é o centro de uma das bases, (Xb, Yb, Zb) é o centro da outra base, e R será o raio do cilindro. Os dois centros definem o eixo do cilindro.
- `cone{ < Xa, Ya, Za >, Ra, < Xb, Yb, Zb >, Rb texture{ Tx } }`
Este comando acrescenta um cone truncado, em posição arbitrária. O ponto (Xa, Ya, Za) é o centro de uma das bases do cone, Ra é o raio dessa base, (Xb, Yb, Zb) é o centro da outra base, e Rb é o raio dessa outra base. Use $Ra > 0$ e $Rb = 0$ para obter um cone inteiro com vértice em (Xb, Yb, Zb) .
- `box{ < Xa, Ya, Za >, < Xb, Yb, Zb > texture{ Tx } }`
Este comando acrescenta uma caixa com lados paralelos aos eixos X , Y e Z . O ponto (Xa, Ya, Za) é um canto qualquer da caixa, e (Xb, Yb, Zb) é o canto oposto. Ou seja, a caixa vai de Xa até Xb na direção X , de Ya até Yb na direção Y , etc..

Em todos os comandos acima, Tx deve ser o nome de uma “tinta” definida previamente com `#declare Tx = texture{ ... }`. Veja o arquivo `main.pov` da aula anterior. Note que não há vírgula antes de `texture` Note também que, em POV-Ray, as coordenadas de pontos se escrevem como `<10,20,30>` e não como `(10,20,30)`.