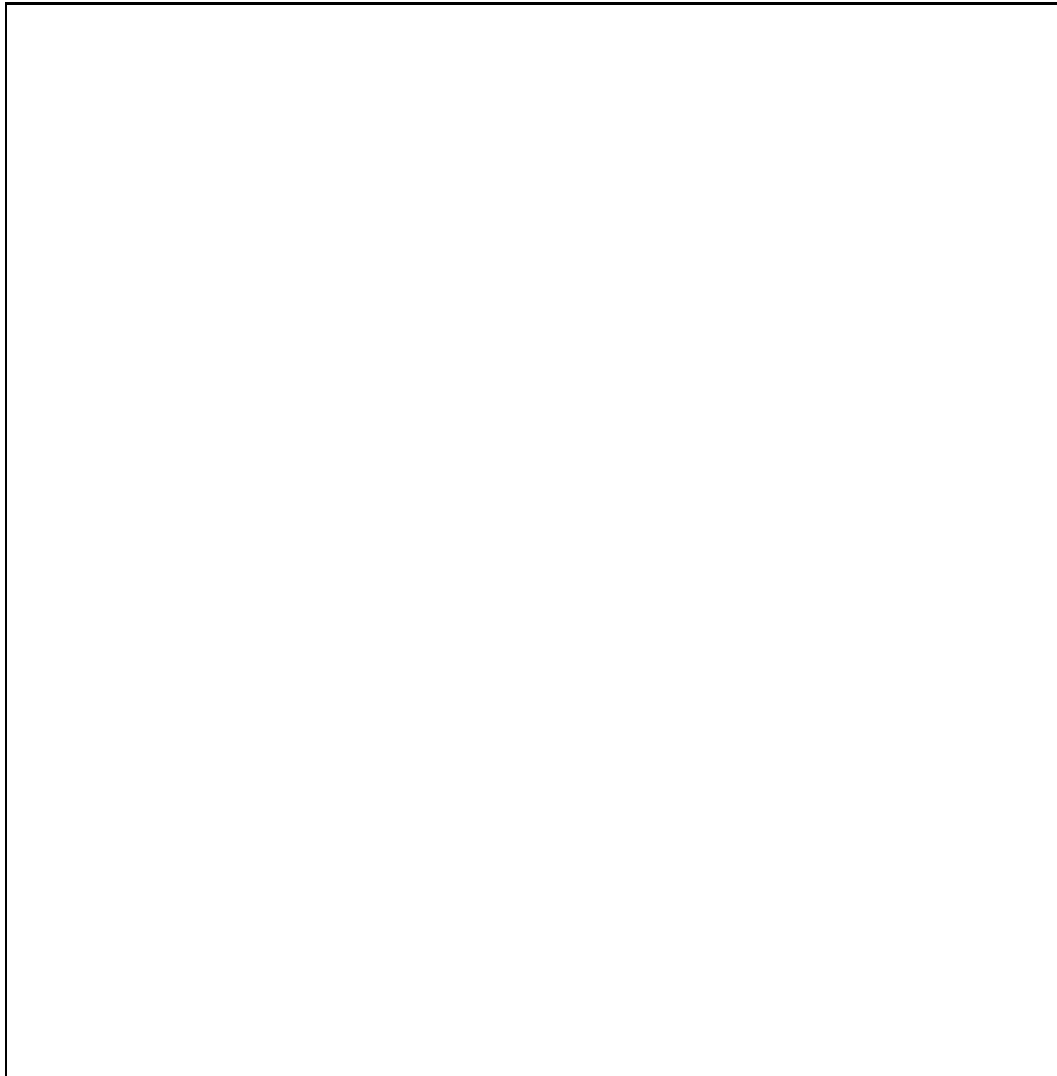


MC930A – Computação Gráfica - 2017-S1 - Jorge Stolfi
Trabalho de laboratório 01
Pen Drive Multifuncional

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

Enunciado. Sua tarefa nesta aula prática é produzir uma ilustração POV-Ray de um *Pen Drive Multifuncional* — que, além de armazenar dados, tem alguma outra função física como abridor de latas, cortador de charutos, descaroador de azeitonas, etc.

Parte 1. Antes de começar a programar, desenhe no espaço abaixo (à mão livre, em perspectiva aproximada) um esboço da sua máquina. **Esta parte deve ser executada e entregue ao professor nos primeiros 15 minutos de aula.**



Parte 2. Produza uma imagem da sua máquina 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**.

Comandos. Os comandos POV-Ray que produzem esses sólidos tem a forma

- `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` do exercício `tp00`. Note que, em POV-Ray, as coordenadas de pontos se escrevem `<...>` e não `(...)`, e não há vírgula antes de `texture`.

Observações. 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, neste exercício ou em exercícios anteriores.

Lembre-se de que todo trabalho prático é **individual**. Você pode pedir ajuda ao colega encontrar erros ou explicar conceitos, mas não para lhe dizer o que escrever. Seu projeto deve ser original, e você mesmo deve bolar os comandos necessários que são objeto da aula!

Não se esqueça de executar o comando `make export` até o final da aula.