

MC930/MO603 – Computação Gráfica - 2003-S1 - Jorge Stolfi  
Trabalho de laboratório 05: Supormercado

Nome	RA	Nota
------	----	------

O objetivo deste exercício é treinar o uso de *comandos de repetição* combinados com *comandos condicionais* e *números aleatórios* na construção de cenas semi-repetitivas.

**Enunciado.** Apesar de tudo o que foi dito e escrito sobre o assunto, as causas da funesta hecatombe dos dot-coms ainda não foram devidamente esclarecidas. Uma teoria recente, sustentada por um conjunto possivelmente não vazio de pesquisadores, é que os sites de venda pela internet nunca entenderam o essencial: o que os consumidores *realmente* procuram ao entrar numa loja *não* são nem produtos ou serviços. Segundo esses luminares, o principal atrativo de um supermercado é o *ambiente*. Ao contemplar as extensas prateleiras abarrotadas de produtos, o freguês se sente “nadando na abundância”: ver todas aquelas coisas na sua frente, ao alcance da mão, é, psicologicamente, quase a mesma coisa do que possuí-las. Em contraste, os sites de vendas pela internet tem apenas a oferecer a visão de uma telinha minúscula (ainda por cima atravancada com frames, ícones, anúncios e logotipos), na qual aparece apenas um produto por vez — se tanto.

Será verdade? Bom, como diria o diabo, tentar não custa. Portanto o exercício de hoje é implementar um revolucionário conceito de e-commerce: um supermercado virtual, ou seja um *supormercado*, onde o cliente pode ver de uma única vez toda a variedade de produtos (reais ou virtuais) que o vendedor tem a oferecer, dispostos num desfile arrebatador de alas, estantes, prateleiras, fileiras, etc..

**Parte 1.** Antes de começar a programar, desenhe no espaço abaixo (à mão livre, em perspectiva aproximada) um esboço indicativo do seu supormercado.

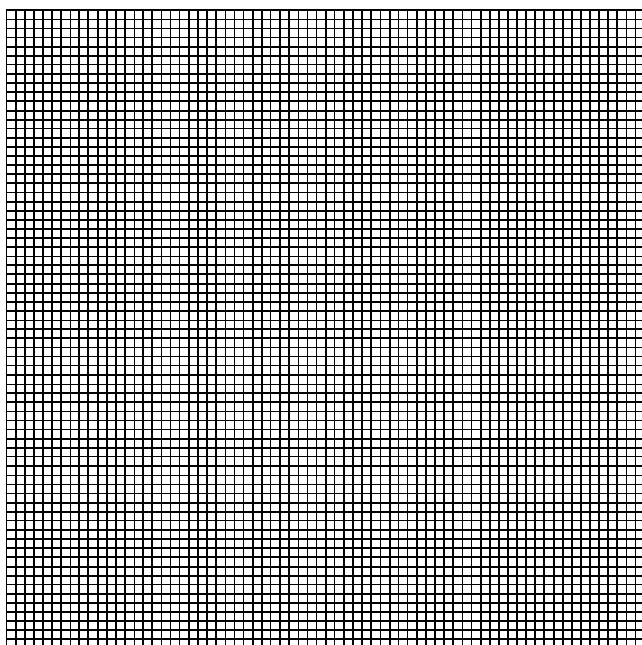
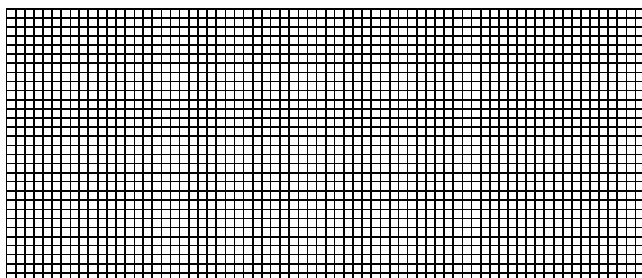


## MC930/MO603 – Computação Gráfica - 2003-S1 - Jorge Stolfi

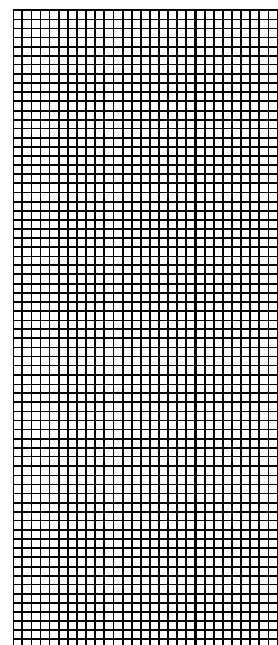
### Trabalho de laboratório 05: Supormercado

**Parte 2.** Use os quadriculados abaixo para planejar a disposição das estantes e prateleiras de seu supormercado.

Frente



Topo



Lado

**Parte 3.** Produza uma imagem de seu supormercado usando POV-ray. O arquivo de descrição deve ser construído manualmente, com um editor de texto comum, **sem** o auxílio de qualquer editor gráfico ou outra ferramenta de modelagem geométrica.

## MC930/MO603 – Computação Gráfica - 2003-S1 - Jorge Stolfi

### Trabalho de laboratório 05: Supormercado

#### Dicas:

- Altere o Makefile e as dimensões da câmera para gerar imagens panorâmicas com proporção 3:1 (p.ex.  $600 \times 200$  ou  $480 \times 160$ ).
- Para repetir um trecho de descrição POV-Ray múltiplas vezes, usa-se o comando `#while ...#end`, conforme o exemplo abaixo:

```
#declare fileira =
union {
  #declare i = 0;
  #while (i < 10)
    #declare r = 1 + 0.1*i;
    #declare treco =
    #if (i = 7)
      box { <-r,-r,-r>, <+r,+r,+r> }
    #else
      sphere{ <0,0,0>, r }
    #end
    object{ treco translate < 3*i, i*i/5, 0 > }
    #declare i = i + 1;
  #end
  pigment{ color rgb <1,0,0> }
}

object{fileira translate <0,0,-3>}
object{fileira translate <0,0,+3>}
```

Os comandos `#while` e `#if` podem ser encaixados à vontade, como em C ou Pascal. Note que os parênteses em volta da condição são obrigatórios, e que o símbolo de igualdade é '=' (e não o '==' do C). Todos os comandos com # são executados numa primeira fase, durante a leitura do arquivo `.pov`. Os demais comandos (incluindo `object`) são interpretados numa segunda fase.

- A função `rand(G)` devolve um número aleatório entre 0 e 1. Para obter números entre 0 e um valor qualquer `vmax`, use `vmax*rand()`. O parâmetro `G` é um “gerador” de números aleatórios, que é criado pela função `seed(A)`. Por sua vez, o parâmetro `A` é um inteiro, de preferência  $> 10000$ , que permite variar os números produzidos pelo gerador. (O mesmo valor de `A` resulta na mesma seqüência de números “aleatórios.”) Veja o exemplo:

```
#declare gerador = seed(123456);
#declare i = 0;
#while(i < 10)
  #declare r = 1 + 0.1*i;
  sphere{ < 3*i, 10*rand(gerador), 0 >, r }
  #declare i = i + 1;
#end
```