

**MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi**  
**Trabalho de laboratório 04: O Trem da Alegria**

|             |           |             |
|-------------|-----------|-------------|
| <b>Nome</b> | <b>RA</b> | <b>Nota</b> |
|-------------|-----------|-------------|

**Objetivos:** treinar (mais um pouco) o uso de comandos iterativos e condicionais, e números aleatórios, na construção de cenas complexas.

**Enunciado:** Com o fim próximo das eleições, todo empreendedor previdente já deve estar se preocupando com o “trem da alegria” que inevitavelmente se seguirá à posse. Ciente de sua missão de formar futuros líderes nacionais, a universidade não pode deixar de dar sua contribuição a essa bem-estabelecida tradição político-ferroviária. Assim, a tarefa de hoje será visualizar o tal trem.

Infelizmente este enunciado não pode ser mais específico, pois as expectativas a respeito desse objeto naturalmente dependem muito de cada pessoa e de suas afiliações políticas. Tudo que podemos dizer com certeza é que envolve dinheiro, uma locomotiva poderosa (com quanto mais poder, melhor), e um monte de vagões que vão aonde ela for. Também sabemos que o trilho é uma curva fechada, de modo que o trem passa na mesma estação a cada quatro anos, sempre no mesmo sentido (carreira acima).

**Parte 1:** Antes de começar a programar, desenhe no espaço abaixo (à mão livre, em perspectiva aproximada) um esboço da sua concepção do trem da alegria e seus trilhos. O trem deve ter uns 5-10 vagões, cada qual com pelo menos 8 rodas e 5 janelas. Tenha em mente que o projeto deve usar os comandos de controle `#while` e `#if` para evitar repetição desnecessária de código, e a função `random`. **Esta parte deve ser executada e entregue ao professor nos primeiros 15 minutos da aula.**

MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi  
Trabalho de laboratório 04: O Trem da Alegria

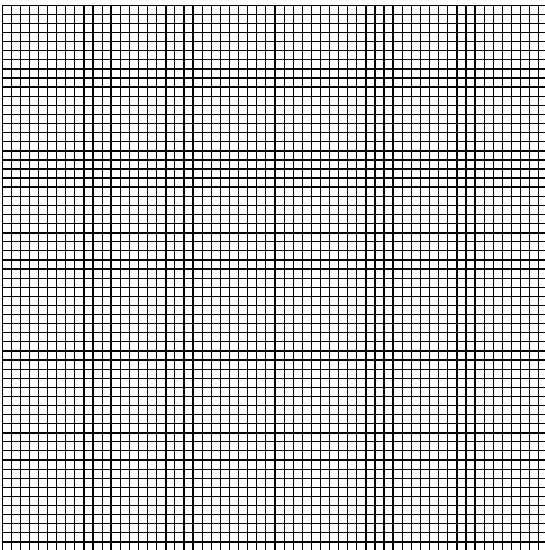
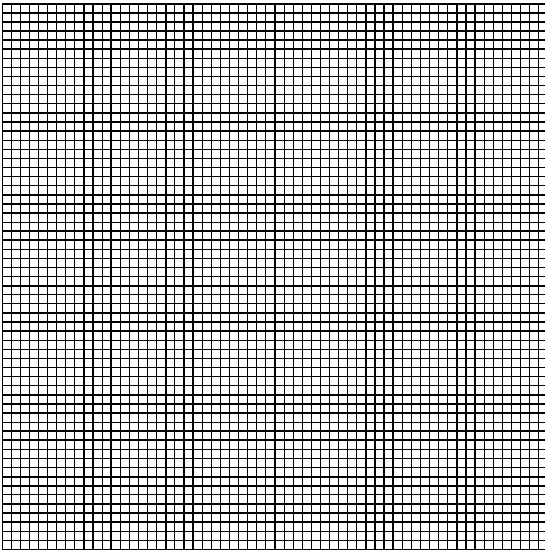
Nome

RA

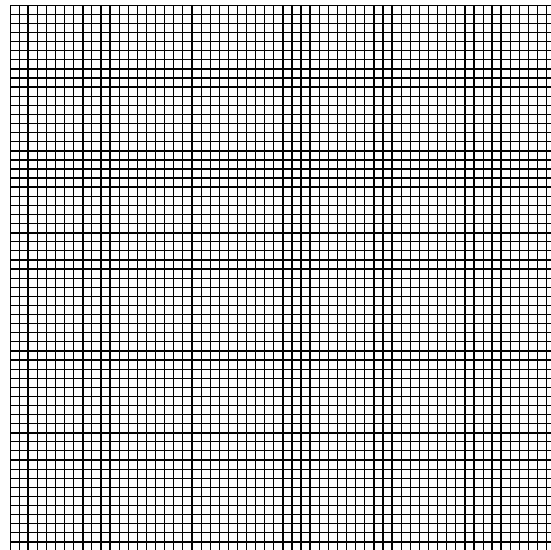
**Parte 2:** Recomendamos usaros quadriculados abaixo para desenhar visões ortogonais da locomotiva e/ou vagões, para facilitar a determinação de coordenadas necessárias para a modelagem.

dos objetos.

Vista de topo



Vista de frente



Vista de lado

## MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi

### Trabalho de laboratório 04: O Trem da Alegria

**Parte 3:** Produza uma imagem do prédio 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. O modelo deve usar comandos `#while` e `#if`, como na semana passada, para construir o modelo de maneira sucinta.

Modifique o Makefile para gerar imagem no formato panorâmico largo com proporções 2:1 (por exemplo 600 pixels de largura e 300 pixels de altura). Modifique a câmera de acordo (`right -1.0` e `up 0.50`) **O resultado desta parte deve ser entregue executando o comando `make export` até o final da aula.**

- 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 “objeto 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
```