

MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi

Trabalho de laboratório 03: O Templo da Computação

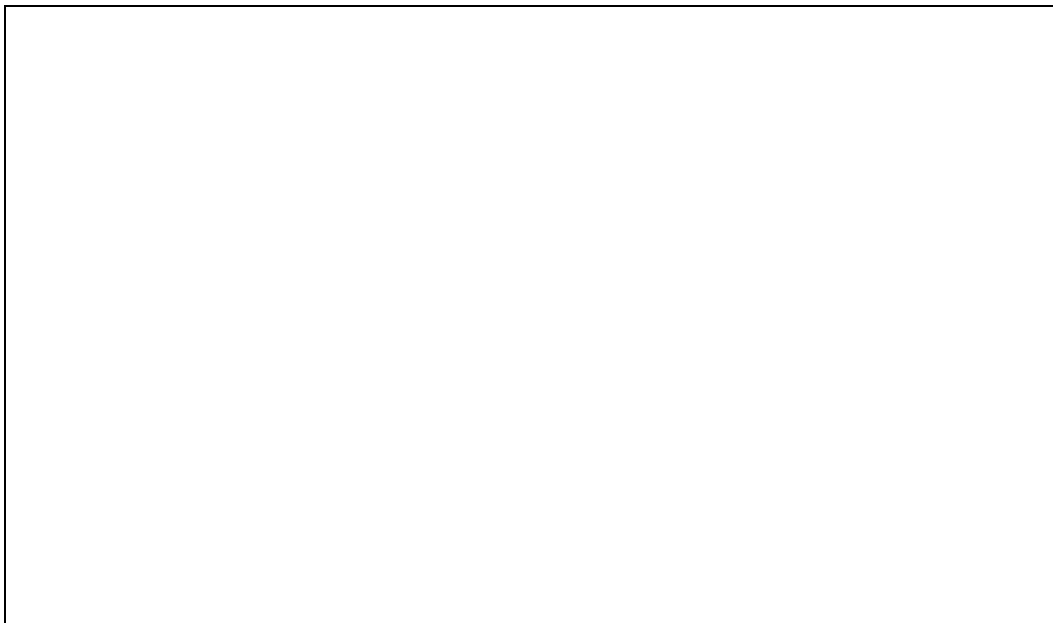
Nome	RA	Nota
------	----	------

Objetivos: treinar o uso de comandos iterativos e condicionais na construção de cenas complexas.

Enunciado: O Instituto de Computação da Unicamp tem planos para construir um prédio próprio, na quadra vaga logo acima do IC-3. Apesar do projeto já estar pronto, ninguém sabe quando teremos dinheiro para a construção. Enquanto isso, nada nos impede de tentar bolar um projeto melhor. Quem sabe, se surgir uma idéia realmente fantástica, o Instituto pode ser convencido a jogar fora as plantas e começar tudo de novo.

Portanto, a tarefa de hoje é produzir um modelo de um prédio com formato adequado para a missão do IC e grande o bastante para conter os egos inflados dos computeiros. Devido ao tempo limitado, o projeto não deve ser muito complicado, mas deve ter pelo menos 4 andares e umas 100 salas, com janelas, e um portal de entrada, e algum tipo de aleatoriedade para quebrar a monotonia da fachada.

Parte 1: Antes de começar a programar, desenhe no espaço abaixo (à mão livre, em perspectiva aproximada) um esboço da sua idéia para o prédio do IC. O projeto deve usar os comandos de controle `#while` e `#if`, e a função `random`. **Esta parte deve ser executada e entregue ao professor nos primeiros 15 minutos da aula.**



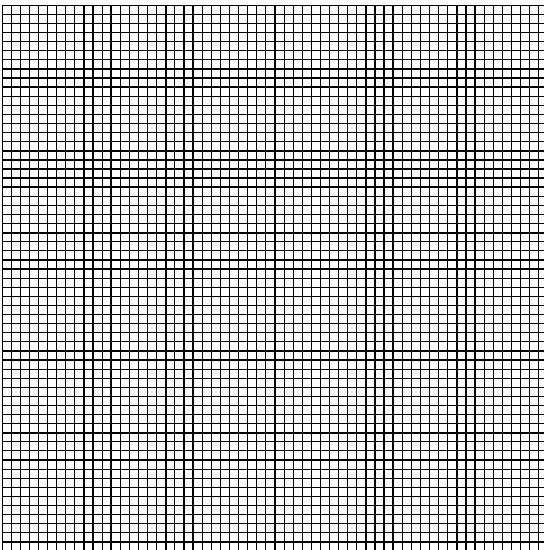
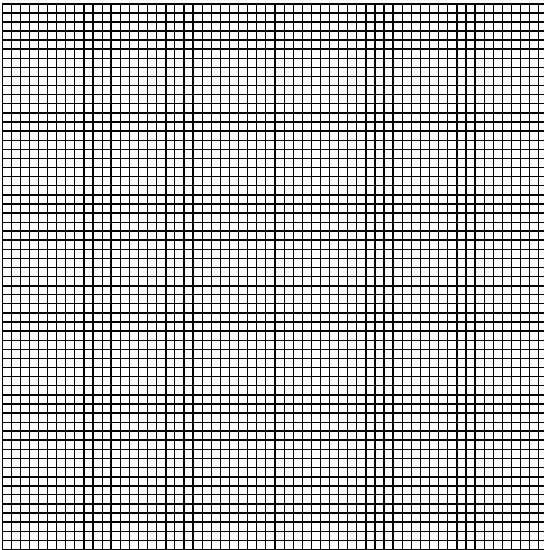
MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi
Trabalho de laboratório 03: O Templo da Computação

Nome

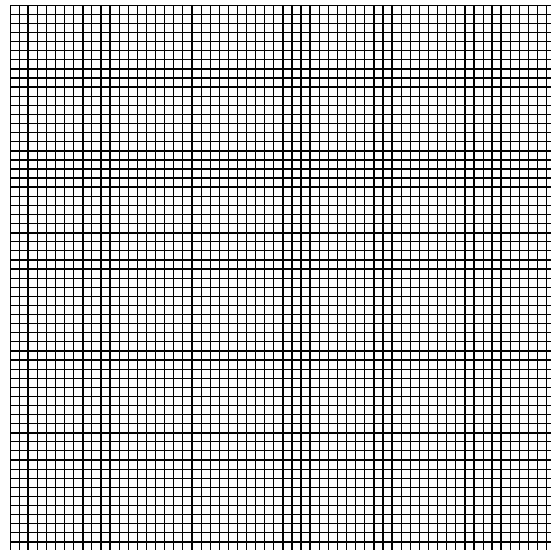
RA

Parte 2: Nos quadriculados abaixo, desenhe duas vistas principais (frente, lado, ou topo) do seu prédio. Estas vistas devem ser projeções ortogonais nos respectivos planos de coordenadas (YZ, XZ, XY), sem perspectiva. Indique os eixos em cada desenho. A finalidade destes desenhos é facilitar a determinação das coordenadas aproximadas dos objetos.

Vista de topo



Vista de frente



Vista de lado

MC930 – Computação Gráfica - 2004-S2 - Jorge Stolfi

Trabalho de laboratório 03: O Templo da Computação

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. Use o formato **horizontal** para a imagem (como no exercício TP01). **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
```