

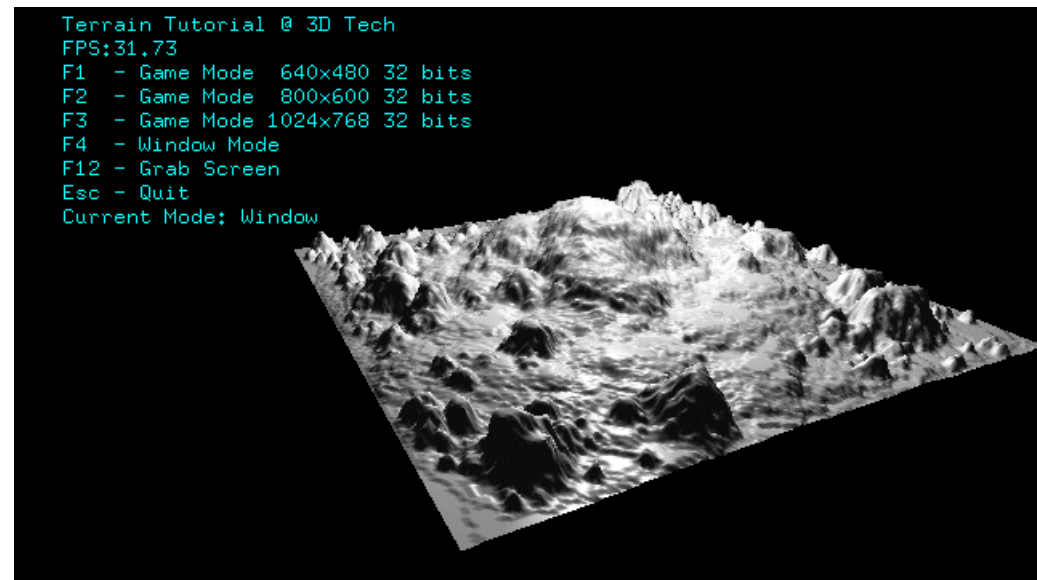
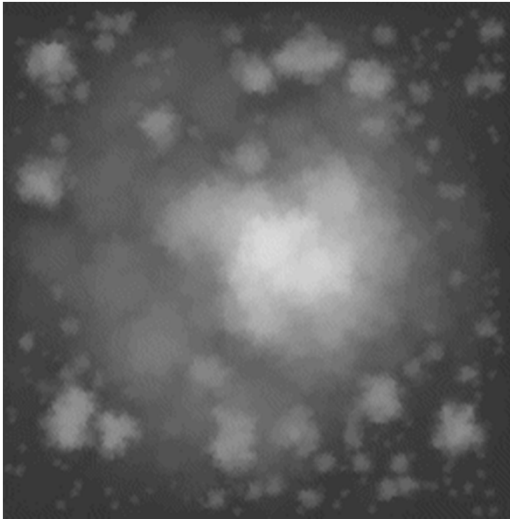


Geração de Geometria - Terrenos



Mapas de Alturas

Intensidade por pixel pode representar uma altura numa grelha regular





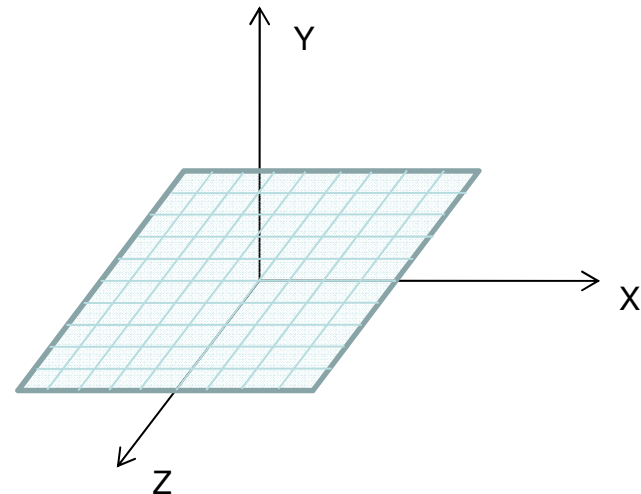
Terrenos a partir de Imagens

- Objectivo:

- Dada uma imagem criar uma grelha regular em que a altura de cada ponto da grelha corresponde à intensidade do pixel correspondente.

- Tarefas:

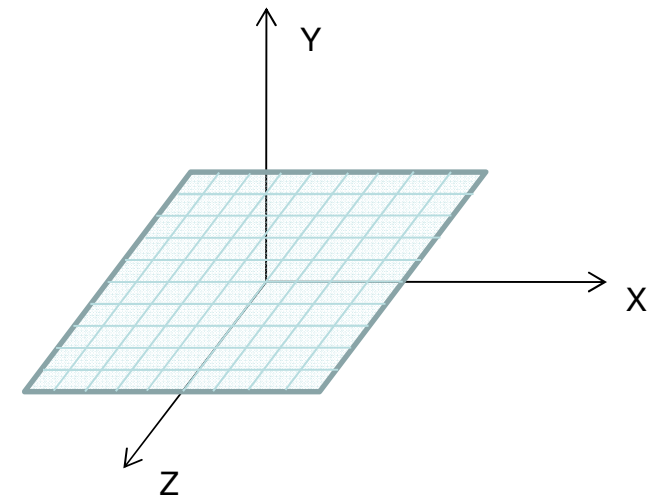
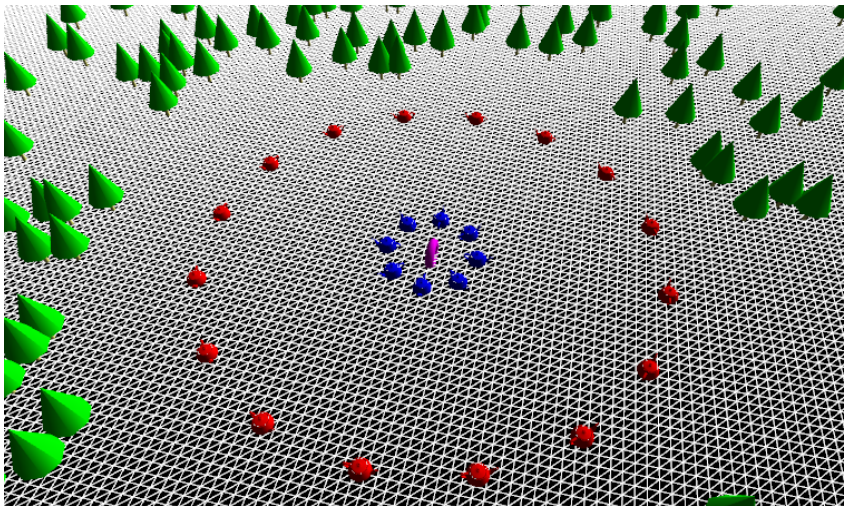
- Carregar a imagem
- Criar a geometria a partir da matriz de pixels extraída da imagem.





Terrenos a partir de Imagens

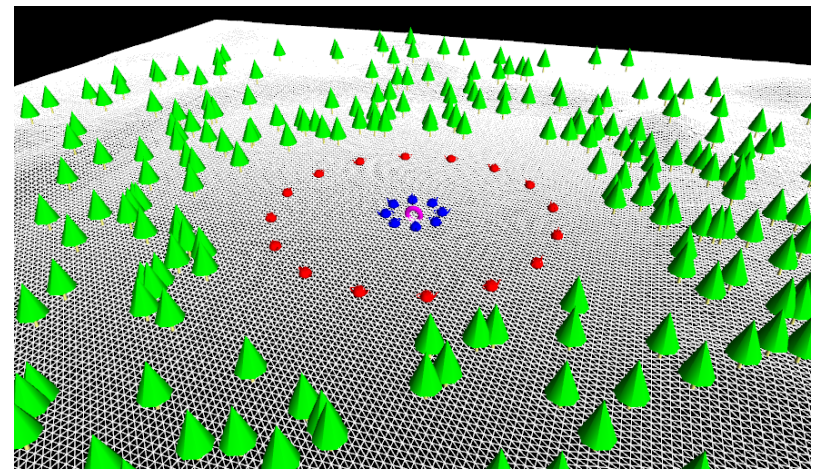
- Primeiro passo:
 - Construir uma grelha regular com altura 0.0 com as dimensões da imagem





Terrenos a partir de Imagens

- Segundo Passo:
 - Atribuir as alturas aos vértices da grelha de acordo com os dados da imagem
 - Construir uma função $h(\text{int } i, \text{int } j)$ que devolva as alturas para os pixels da imagem (necessária para a definição da geometria do terreno)
 - Construir uma função $hf(\text{float } i, \text{float } j)$ que devolva as alturas para todos os pontos do terreno (necessário para colocar as árvores na altura certa)





DevIL

- Converter para escala de cinzentos (mapa de alturas)

```
ilConvertImage(IL_LUMINANCE,IL_UNSIGNED_BYTE);
```



DevIL - Carregar uma imagem

- Exemplo para carregar uma imagem:

```
#include <IL/il.h>
```

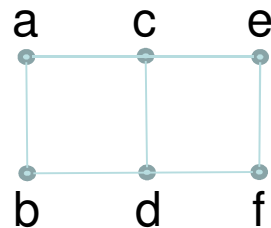
```
...
```

```
unsigned int t, tw, th;  
unsigned char *imageData;  
ilGenImages(1, &t);  
ilBindImage(t);  
ilLoadImage((ILstring) "terreno2.jpg");  
tw = ilGetInteger(IL_IMAGE_WIDTH);  
th = ilGetInteger(IL_IMAGE_HEIGHT);  
ilConvertImage(IL_LUMINANCE, IL_UNSIGNED_BYTE);  
imageData = ilGetData();
```



Triangle Strips

- Definição de tiras de triângulos (triangle strips)
 - Array de triângulos: {a,b,c,d,e,f}
 - `glDrawArrays(GL_TRIANGLE_STRIP, first, count)`
 - Número de tiras = altura da imagem - 1





Geração de terrenos

- Inicialização:
 - Carregar a imagem
 - Criar um array com a geometria do terreno com as alturas correspondentes do terreno
 - Criar VBO e passar array para o OpenGL
- Render:
 - Semântica e desenho do VBO para o terreno (`glDrawArrays`)
 - Desenhar arvores + teapots



Exercício

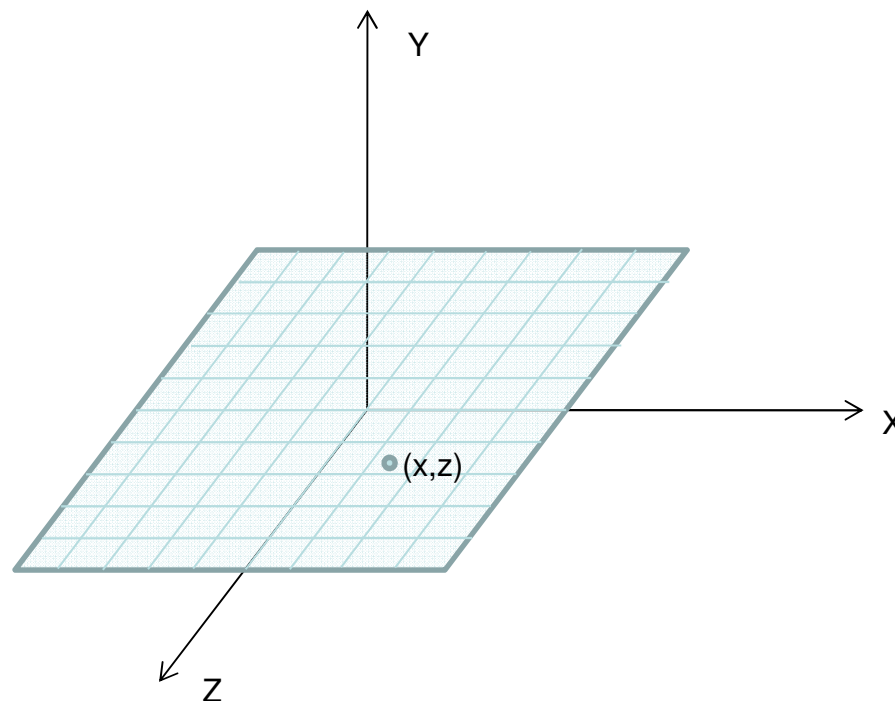
- Dada uma imagem, interpretá-la como um mapa de alturas e gerar a geometria correspondente
- Colocar as árvores na altura correcta
- TPC: Implementar um esquema de *surface following*.



Surface Following

- Problema: determinar a altura do ponto (x,z) de uma grelha.

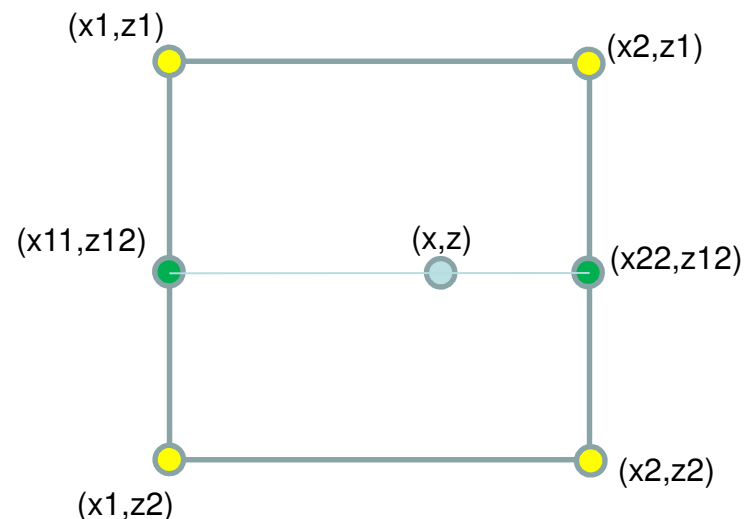
Sendo $h(i,j)$ a função que determina a altura nos vértices da grelha, é necessário determinar a altura do ponto (x,z) a partir dos cantos da célula onde o ponto se encontra





Surface Following

- Através da função h temos acesso às alturas dos cantos da célula (pontos amarelos).
- a altura de $(x11,z12)$ obtem-se por interpolação linear das alturas de $(x1,z1)$ e $(x1,z2)$. Processo semelhante para determinar a altura de $(x22,z12)$.
 - seja fz a parte fraccionária de z :
 - $fz = z - z1; \quad // \quad 0 \leq fz \leq 1$
 - $alt(x11,z12) = h(x1,z1) * (1-fz) + h(x1,z2) * fz$
 - $alt(x22,z12) = h(x2,z1) * (1-fz) + h(x2,z2) * fz$
- A altura de (x,z) obtem-se por interpolação linear entre as alturas de $(x11,z12)$ e $(x22,z12)$
 - $alt(x,z) = alt(x11,z12) * (1 - fx) + alt(x22,z12) * fx$





DevIL

Developers Image Library - Instalar

- Mover a pasta IL para a pasta dos includes
- Mover devil.lib para a pasta das libs
 - Ir a propriedades do projecto;
 - Linker -> Input;
 - Adicionar devil.lib à lista de bibliotecas
- Adicionar devil.dll à pasta das dlls

No código:

- `#include <IL/il.h>`
- A seguir à inicialização e registo dos callbacks do GLUT
 - `ilInit();`