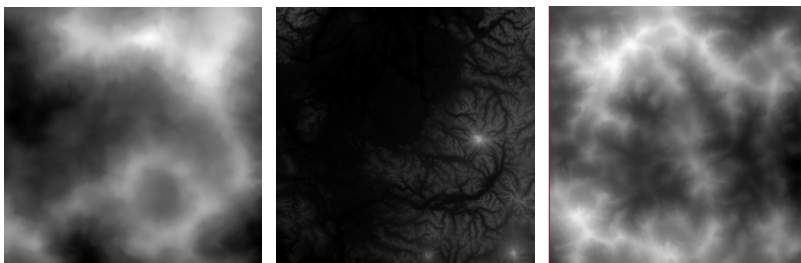


PL 06- Terrain

Notas para a componente prática de Computação Gráfica
Universidade do Minho
António Ramires

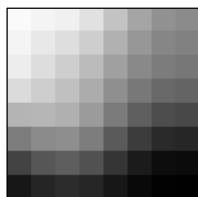
1. Introdução

Com este guião pretende-se explorar a criação de um terreno a partir de mapas de alturas. Para o mapa de alturas podemos usar imagens como as seguintes:



Os pixels destas imagens são interpretados como valores representativos da altura do terreno, sendo branco (255) a altura máxima, e preto (0) a altura mínima. Podemos adicionar um factor de escala para obter um terreno com a gama de alturas desejadas. [\(uma gama de alturas arbitrária?\)](#)

Consideremos uma pequena imagem 8x8:



Se considerarmos que a imagem só tem um canal para cada pixel que representa a intensidade do pixel, os valores dos pixels presentes nesta imagem são:

Commented [CB1]: Imagens muito fixes! Talvez acrescentar uns links para irem buscar estas imagens em full-res? 😊

Formatted: Portuguese (Portugal)

251	245	241	226	196	166	146	140
245	233	223	207	178	152	135	130
238	222	207	188	161	138	124	119
220	207	193	173	144	120	105	100
181	183	177	156	123	95	77	72
125	140	143	125	90	60	42	38
67	86	94	81	53	27	13	11
23	37	44	38	23	9	2	1

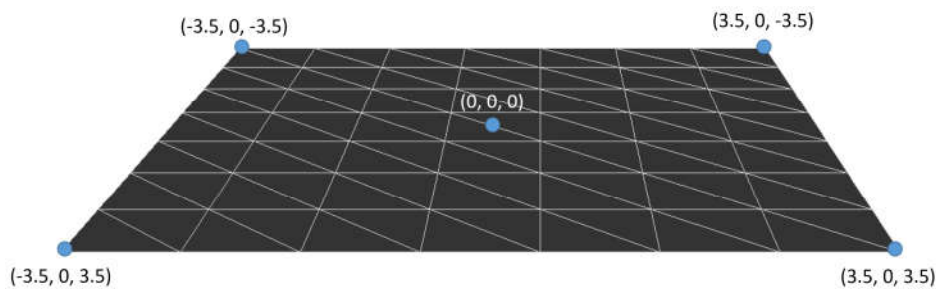
O primeiro passo será, portanto, carregar esta imagem e guardar ~~aos~~os valores dos pixels numa ~~variável~~(em memória? Para sugerir um array). Em seguida temos de criar uma grelha. Os valores dos pixels serão utilizados para a componente Y de uma grelha assente no plano XZ.

2. Criar uma grelha

Neste caso a nossa imagem tem a dimensão 8x8, e a nossa grelha terá a mesma dimensão. A figura seguinte ilustra a forma da grelha, ainda sem considerar as alturas.

A grelha tem 8 vértices por 8 vértices e está no plano XZ. Se considerarmos um espaçamento entre vértices de 1 unidade e a grelha centrada na origem teremos os valores de X e Z a variar entre -3.5 e 3.5.

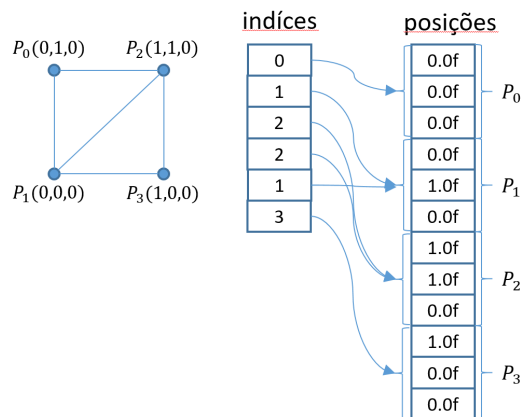
A imagem seguinte mostra a grelha e as coordenadas dos vértices ~~de~~nos extremos da grelha e da origem (notem que a origem não corresponde a nenhum vértice da grelha).



Para a construção da grelha iremos usar *triangle strips* (*GL_TRIANGLE_STRIP*) em vez da habitual lista de triângulos. Uma strip de triângulos é uma ~~file-fita~~ de triângulos, sendo que o terreno pode ser visto como uma coleção de N-1 ~~file-fitas~~, sendo N=8 neste caso. Cada ~~file-fita~~ é constituída por duas linhas adjacentes de vértices, sendo a ordem dos vértices ~~relevante, como sempre (embora agora com uma ordem específica)~~.

A diferença para a lista de triângulos que temos usado até agora é a forma como submetemos os vértices e, consequentemente, como o OpenGL os interpreta na altura de desenho.

Consideremos um exemplo mínimo: um quadrado formado por dois triângulos. Podemos criar um array de coordenadas e outro de índices tal como no guião da aula 4:

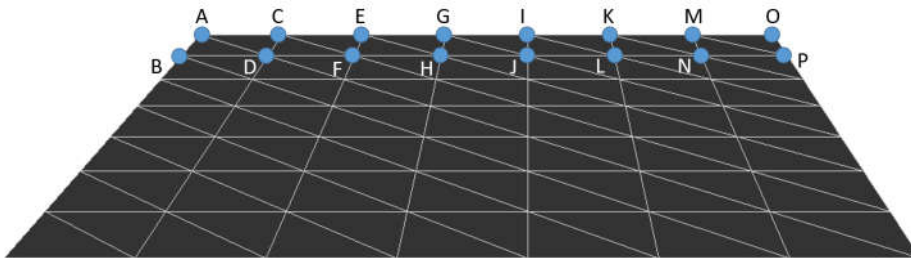


Desenhar com strips implica definir os vértices no array de posições pela seguinte ordem: P_0, P_1, P_2, P_3 , tal como na figura acima. Os vértices de uma fila de triângulos são passados por colunas. A primeira coluna tem os vértices P_0, P_1 , e a segunda coluna tem os vértices P_2, P_3 . No fundo é como se o array de índices fosse implícito, não sendo necessário criá-lo. Ao especificar que queremos desenhar *strips*, com `GL_TRIANGLE_STRIP`, o próprio OpenGL criará o array de índices.

Regressando à nossa grelha, a imagem seguinte apresenta de novo a grelha, atribuindo uma letra a cada vértice da primeira fila de triângulos.

Commented [CB2]: Maybe too snarky

Commented [AR3R2]:



O array de vértices para desenhar a strip terá os vértices por ordem alfabética, ou seja:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Notem que cada letra representa as três coordenadas do vértice associado, ou seja o nosso array com valores das coordenadas será:

-3.5	0.0	-3.5	-3.5	0.0	-2.5	-2.5	0.0	-3.5	-2.5	0	-2.5	...	3.5	0.0	-3.5	3.5	0.0	-2.5
A		B		C		D		O		P								

Tal como mencionado anteriormente, cada par de linhas consecutivas de vértices forma uma strip, o que resulta em N-1 strips independentes. A definição de strips implica a repetição de vértices, por exemplo os vértices da segunda linha fazem parte da primeira strip, como definido acima, e também da strip composta pela fila de triângulos definida pela segunda e terceira linhas de vértices. De qualquer forma esta repetição é muito menor que a necessária desenhando triângulos com GL_TRIANGLES.

A título de exercício, considerando filas de triângulos com N triângulos, determine o limite da razão, quando N tende para números grandes, entre o número necessário de vértices que o array de posições terá de ter considerando GL_TRIANGLES (sem índices) e GL_TRIANGLE_STRIP.

As strips são independentes, devendo ser desenhadas uma a uma. Isto implica uma de duas alternativas:

- ou criar um só VBO com as strips consecutivas e desenhar cada strip independentemente com um offset no array;

- ou criar N-1 VBOs e desenhá-los independentemente.

-A segunda opção implicar seguir a metodologia que está definida no guião 4 para cada strip.

A primeira opção, um único VBO para todas as strips, poderá ser desenhada tendo em conta que sabemos a dimensão da imagem. O número de strips é a altura da imagem menos 1 ($8-1=7$ no nosso caso), e a largura da imagem corresponde ao número de vértices por fila (8). Sabemos também que cada strip tem o dobro dos vértices que constituem uma fila de vértices (neste caso $8 \times 2 = 16$).

Para desenhar a strip i temos de desenhar 16 vértices, sendo o offset da strip i , $16 \times i$. Considerando que os valores da altura e largura da imagem estão armazenados nas variáveis `imageWidth` e `imageHeight`, poderíamos utilizar o código que se segue para desenhar as strips:

Commented [CB4]: Acho que precisa de clarificação. Percebo o essencial, relacionar o número de vértices precisos com GL_TRIANGLES e GL_TRIANGLE_STRIP. Mas confesso que não entendi a expressão, "limite da razão". Talvez "considerando filas de triângulos com N triângulos subdivisões"?

Commented [CB5]: Se for para avaliação, talvez devesse estar nomeado como exercício 1 ou algo assim

Commented [CB6]: Ou apenas um VBO com triângulos degenerados, mas esta é capaz de lhes fritar os miolos 😊

```
glBindBuffer(GL_ARRAY_BUFFER, buffers[0]);
glVertexPointer(3, GL_FLOAT, 0, 0);

for (int i = 0; i < imageHeight - 1 ; i++) {
    glDrawArrays(GL_TRIANGLE_STRIP, (imageWidth) * 2 * i, (imageWidth) * 2);
}
```

Formatted: Portuguese (Portugal)

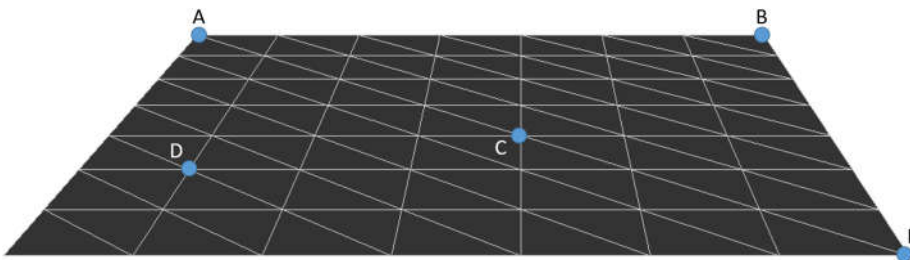
Formatted: Portuguese (Portugal)

Notem que as instruções `glBindBuffer` e `glVertexPointer` só são executadas uma vez. Em `glDrawArrays` usamos o segundo parâmetro para indicar o offset dentro do array (em número de vértices), e o terceiro para indicar o número de vértices a desenhar para uma strip. O primeiro parâmetro ~~especifica~~especifica que cada secção do VBO activo contém uma fila de triângulos, ou seja uma `TRIANGLE_STRIP`.

3. Atribuição de alturas à grelha

Neste momento já temos a grelha desenhada considerando a coordenada Y de todos os vértices igual a 0. Falta agora atribuir um valor à componente Y com base nos valores dos pixels da imagem.

A imagem é composta por uma matriz de pixels, sendo que cada pixel só tem um canal que representa o valor da intensidade do pixel. Consideremos alguns pontos da grelha e a matriz que contém os valores dos pixels da imagem:



^A 251	245	241	226	196	166	146	^B 140
245	233	223	207	178	152	135	130
238	222	207	188	161	138	124	119
220	207	193	173	144	120	105	100
181	183	177	156	^C 123	95	77	72
125	^D 140	143	125	90	60	42	38
67	86	94	81	53	27	13	11
23	37	44	38	23	9	2	^E 1

As letras na matriz da imagem estão nos pixels que teremos de ler para atribuir as alturas ao respectivo vértice na grelha.

Um pormenor importante é o facto de ~~ao aceder aos pixels da imagem o resultado não virão estarem organizados~~ numa matriz, mas sim num array linear. Ou seja, ao ~~aceder-carregar~~ aos dados da imagem o resultado será um array linear com a seguinte forma:

251	245	241	226	196	166	146	140	245	233	223	207	178	152	135	130	...	23	37	44	38	23	9	2	1
Linha 1								Linha 2								Linha 8								

Para definir a altura de cada vértice, este array deve ser indexado, fazendo a transformação adequada de duas coordenadas (x,z) para uma única coordenada correspondente ao índice no array.