

## Displace2 (displace2.\*)

2.5 punts

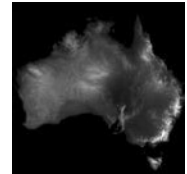
Volem implementar una versió de **displacement mapping** basada en el VS.

El VS serà el responsable d'aplicar un desplaçament (en direcció de la normal) a tots els vèrtexs. Per dur aquesta tasca, el VS farà servir un sampler amb el height map (del qual usarem la component R),

```
uniform sampler2D heightMap;
```

i un float amb l'escala del desplaçament a aplicar,

```
uniform float scale = 0.05;
```



Aquest exercici està pensat per l'objecte **plane256.obj**, que està molt subdividit i té un nombre elevat de vèrtexs. Aquest objecte no té coordenades de textura adjacents (les volem entre 0 i 1); per això, no feu servir texCoord, sinó unes coordenades de textura adhoc:

```
vec2 st = 0.49 * vertex.xy + vec2(0.5);
```

Segui  $r$  el valor de la component vermella del heightfield a les coordenades  $st$ . La magnitud del desplaçament a aplicar serà  $scale * r$ . El VS l'aplicarà, en **object space**, en direcció de la normal.

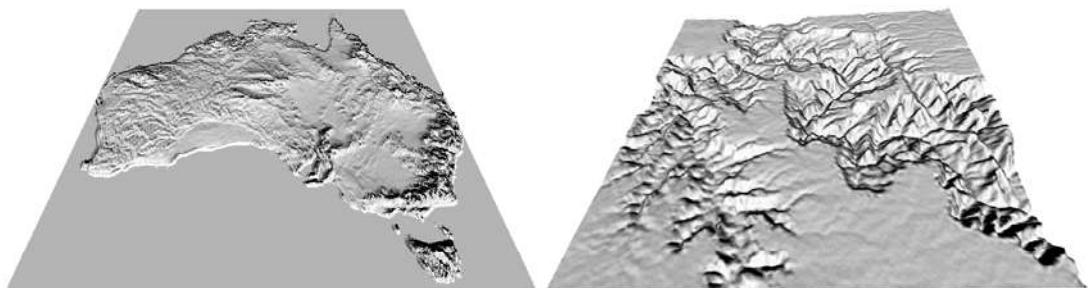
El FS calcularà una normal per fragment com segueix. Primer, farà tres accessos a la textura del heightfield per calcular-ne el gradient  $G$  al punt  $st$  que rebrà del VS. Per calcular el gradient, feu servir el mètode de diferències avançades (*forward differences*), amb un  $\epsilon = 1.0/128$  per avançar les coordenades  $st$ .

Un cop calculat  $G$ , la normal del fragment en *object space* la calculeu com el resultat de normalitzar el vector  $(-G.x, -G.y, smoothness)$ , on  $smoothness$  és un uniform,

```
uniform float smoothness = 25.0
```

Un cop passeu aquesta normal a *eye space*, el color final del fragment serà  $vec4(N.z)$ .

Aquí teniu el resultat (**plane256.obj**) amb diferents height fields i valors d'escala:



### Identificadors (ús obligatori):

```
displace2.vert  displace2.frag  
uniform sampler2D heightMap;  
uniform float scale = 0.05;  
uniform float smoothness = 25.0;
```