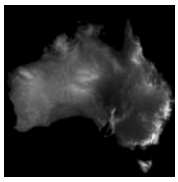
Displace (displace.*) 2.5 punts

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;
```

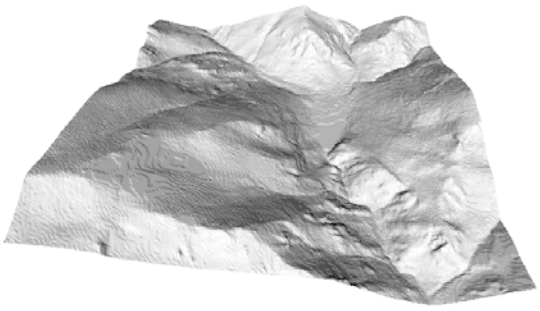
Aquest exercici està pensat per l'objecte **plane256.obj**, que està molt subdividit i té un nombre elevat de vèrtexs. Aquest objecte no té coordenades de textura adients (les volem entre 0 i 1); per això, no feu servir texCoord, sinó unes coordenades de textura adhoc:

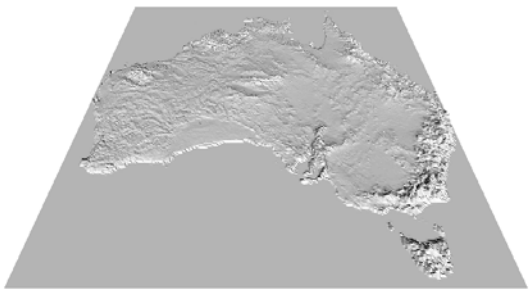
```
vec2 st = 0.49 * vertex.xy + vec2(0.5);
```

Sigui 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 haurà de calcular el color (escala de grisos) de manera similar a l'exercici de càlcul de la normal en el fragment shader. El VS haurà de passar el punt desplaçat de object space a eye space, i enviar-li al FS. El FS usarà aquest punt per calcular (amb dFdx i dFdy) una normal N unitària, i assignarà com a color del fragment vec4(N.z).

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





Identificadors (ús obligatori):

```
displace.vert  displace.frag
uniform sampler2D heightMap;
uniform float scale = 0.05;
```

XRays (xrays.*) 2.5 punts

Escriu un VS i un FS per simular una mena de lupa, controlada amb el mouse, que permeti veure les capes interiors d'un dibuix d'anatomia. Farem servir quatre textures (foot0.jpg ... foot3.jpg):

```
uniform sampler2D foot0;
uniform sampler2D foot1;
uniform sampler2D foot2;
uniform sampler2D foot3;
```

El VS farà les tasques per defecte, però aplicarà un escalat S(0.5, 1, 1) al vèrtex (abans de passar-lo a clip space), de forma que l'objecte **plane.obj** passi a ser rectangular.

Pel FS, us proporcionem un **xrays.frag** que heu de completar. El que ha de fer el FS és:

1. Calcular la distància d (en píxels) del fragment a les coordenades actuals del mouse. Feu servir obligatòriament la funció mouse() que us proporcionem, que retorna les coordenades del mouse en window space.

2. Usar les coordenades de textura habituals per accedir a la textura **foot0** (pell). Sigui C el color resultant. Si **d >= R** (R és una constant que ja teniu declarada al exemple), el color del fragment serà directament C. Altrament, el color final es calcula com segueix.

3. Accedir a la textura indicada pel **uniform int layer=1** per obtenir un altre color D. Per exemple, si layer = 1, cal obtenir el color de la textura foot1. Podeu assumir que layer sempre tindrà valor 0, 1, 2 o 3.

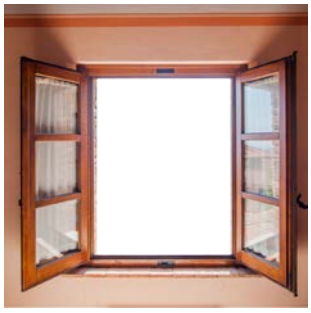
4. El color final del fragment (cas d<R) serà el resultat de fer la interpolació lineal entre D i C, on el paràmetre d'interpolació lineal serà d/R (és a dir, la distància al mouse normalitzada per R). D'aquesta manera el centre del cercle al voltant del mouse mostrarà el color D de la capa interior (indicada per layer) mentre que a mesura que ens allunyem de la posició del mouse es mostrarà gradualment el color C de pell.

Identificadors (ús obligatori):

```
xrays.vert, xrays.frag
uniform sampler2D foot0, foot1, foot2, foot3;
uniform int layer;
```

Magic window (magic.*) 2.5 punts

Escriu un **VS** i un **FS** per tal de simular una finestra a través de la qual es pot veure l'interior d'una habitació, i un altre cop l'exterior. Al ZIP de l'enunciat (i també a /assig/...) trobareu les textures **window.png**, **interior.png** i **exterior.png**:


El VS, a banda de les tasques habituals, li passarà al FS **la normal N en eye space.**

El FS accedirà a la textura **window.png** (amb les coordenades de textura habituals) per obtenir un color que li direm **C**.

Si la component alfa de C és 1.0 (part opaca de la primera finestra), el color del fragment serà C.

Si la component alfa de C és inferior a 1.0, per calcular el color del fragment s'accedirà a la textura **interior.png** (amb coordenades de textura **vtexCoord+0.5*N.xy**) per obtenir un color que li direm **D**.

Si la component alfa de D és 1.0 (part opaca de la finestra interior), el color del fragment serà D. Altrament, el color del fragment serà el color de la textura **exterior.png** al punt de coordenades **vtexCoord+0.7*N.xy**.

Observeu que estem usant un offset en les coordenades de textura que depèn de les components de la normal en eye space. Degut a aquest offset, la part visible de l'interior i de l'exterior depèn de l'orientació del model. Aquí teniu el resultat esperat (**plane.obj**), des de diferents punts de vista:



Identificadors (ús obligatori):

```
magic.vert, magic.frag
uniform sampler2D window;
uniform sampler2D interior1;  // observeu el digit 1 al final
uniform sampler2D exterior2;  // observeu el digit 2 al final
```

Gràfics
Curs 2016/17Q1

Crèdit imatges: Platja: https://www.pexels.com/photo/sea-sky-beach-holiday-878/

Bats (bats.*) 2.5 punts

Escriu un VS+FS que mostrin una sèrie de ratpenats movent-se per la part de la finestra OpenGL ocupada pel model. Farem servir dues textures (bat0.png, bat1.png) amb les ales orientades de forma diferent:

```
uniform sampler2D bat0;

uniform sampler2D bat1;
```

El VS serà el que teniu per defecte.

El FS no farà servir les coordenades de textura habituals, sinó que usarà les coordenades (x,y) del fragment, dividides per 64. És a dir, cada instància del ratpenat ocuparà un espai de 64x64 pixels. A aquestes coordenades de textura li restareu time, de forma que els ratpenats es moguin en diagonal.

Anirem alternant entre les dues textures (bat0 i bat1) dos cops cada dècima de segon, començant per bat0: quan time estigui en [0, 0.05) farem servir bat0; quan time estigui en [0.05, 0.1) farem servir bat1, i així successivament.

Accedireu a la textura que toqui per obtenir un color C. Si la component alfa de C és prou gran (> 0.2), C correspon a la part opaca del ratpenat i el color final del fragment serà negre. Altrament, el color serà frontColor.

Identificadors (ús obligatori):

```
bats.vert, bats.frag
uniform sampler2D bat0, bat1;
uniform float time;
```