

Geometry Image (geometry-image.*)

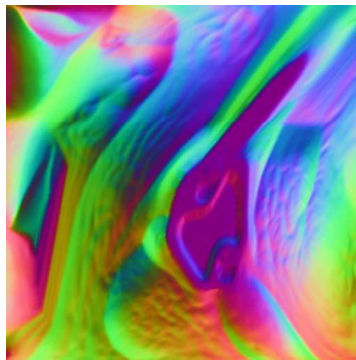
Ureu: ~/assig/grau-g/Viewer/GlarenaSL

El test és només orientatiu

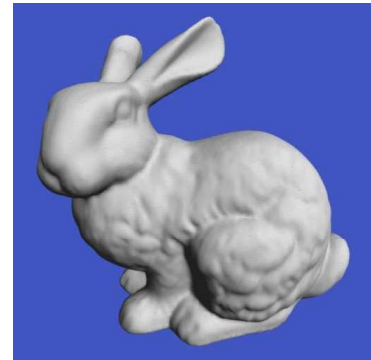
Si tenim una superfície parametritzada, es pot construir una textura on cada texel amb coordenades (s,t) representi la posició (assumirem en *object space*) del punt de la superfície que té aquelles coordenades de textura. La imatge resultant és una *geometry image*, la qual sovint s'utilitza amb una altra imatge que codifica la normal (també en *object space*). Aquí teniu un exemple per a una determinada parametrització del Stanford bunny:



Geometry image



Normal image



Bunny

Escriu un VS+FS que, a partir d'un pla força subdividit (objecte **plane256.obj**), dibuixi la superfície representada per una *geometry image* (**bunny-geo.png**) amb les normals d'una *normal image* (**bunny-norm.png**).

Concretament, el VS farà les següents tasques. Primer, calcularà unes coordenades de textura (s,t) a partir de les coordenades (x,y) del vèrtex. A l'objecte **plane256.obj**, (x,y) varien en [-1,1]; escaleu i traslladeu aquestes coordenades per generar coordenades de textura en [0.004, 0.996].

Useu aquestes coordenades (s,t) per prendre una mostra de les dues textures:

```
uniform sampler2D positionMap;  
uniform sampler2D normalMap1;    // observa el dígit 1 al final
```

Els valors (r,g,b) de la primera textura els interpretarem directament com la nova posició P del vèrtex, en object space. Per tant reconstruirem un objecte amb coordenades en [0,1].

Els valors (r,g,b) de la segona textura codifiquen la normal N del vèrtex, també en object space. Però atès que la textura us retorna valors en [0,1] caldrà fer la transformació adient per tenir les components de la normal en [-1, 1].

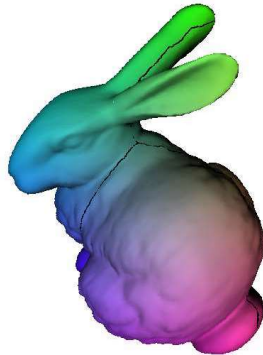
El color del vèrtex dependrà d'un **uniform int mode = 0**. Si **mode** és 0 [4 punts], el color serà directament la posició P del vèrtex en object space:



Ignoreu la discontinuïtat al voltant del cap i les orelles; la superfície no tancarà perfectament.

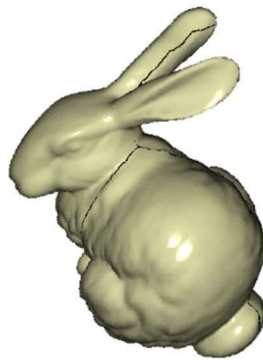
Per la resta de modes, usarem la normal N que heu obtingut de la segona textura, però en eye space i normalitzada.

Si **mode és 1 [2 punts]**, el color serà el mateix que en mode=0, però multiplicat per la $N.z$.

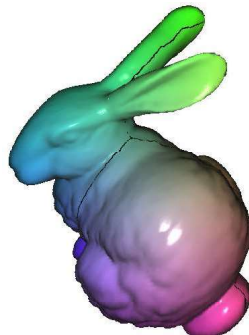


Si **mode és 2 [3 punts]**, el color es calcularà (per vèrtex) amb el model d'il·luminació de Phong,

$$K_a I_a + K_d I_d (N \cdot L) + K_s I_s (R \cdot V)^s$$



Finalment, si **mode és 3 [1 punt]**, també calculareu el color amb el model de Phong, però aquest cop usareu la posició P en substitució de matDiffuse .



El **FS** farà les tasques per defecte.

Identificadors obligatoris:

`geometry-image.vert`, `geometry-image.frag` (Has escrit `geometry-image` correctament? En minúscules?)

Tots els uniforms de l'enunciat, i els requerits per aplicar Phong (`matAmbient`, `matDiffuse`...)