

Beach (beach.*)

Useu **GLarenaSL** per a desenvolupar aquests shaders (/assig/grau-g/Viewer/GLarenaSL).

Escriu un **VS** i un **FS** per tal de simular una finestra a través de la qual es pot veure una palmera moguda pel vent, i un fons. Usarem les textures **interior.png**, **palm-tree2.png** i **dunes.jpg**:



La composició que volem obtenir (amb l'objecte **plane.obj**) la teniu a la dreta.

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

El FS usará tres textures, que haureu de declarar així:

```
uniform sampler2D window;          // interior
uniform sampler2D palm1;           // palm-tree
uniform sampler2D background2;    // dunes
```

Primer accedirà a la textura **window** (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 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 **palm1** amb coordenades de textura $\text{vtxCoord} + 0.25 * \mathbf{N.xy} + \text{vec2}(0.1 * \sin(2 * \text{time}) * \text{vtxCoord.t}, 0)$ per obtenir un color que li direm **D**.

Si la component alfa de **D** és superior o igual a 0.5 (part opaca de la palmera), el color del fragment serà **D**. Altrament, el color del fragment serà el color de la textura **background2** al punt $\text{vtxCoord} + 0.5 * \mathbf{N.xy}$.

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 la palmera i del fons dependrà de l'orientació del model. En el cas de la palmera, també hi ha un desplaçament horitzontal que depèn del temps, per simular l'ondulació produïda pel vent.

Identificadors (ús obligatori):

beach.vert, beach.frag (*minúscules!*)

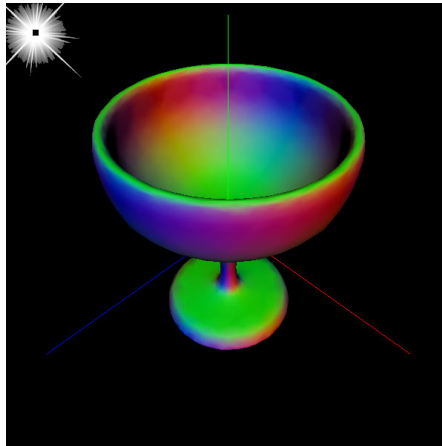
```
uniform float time;
uniform sampler2D window;
uniform sampler2D palm1;          // observeu el dígit 1 al final
uniform sampler2D background2;    // observeu el dígit 2 al final
```



Show light (show-light.*)

Useu **GLarenaSL** per a desenvolupar aquests shaders (/assig/grau-g/Viewer/GLarenaSL).

Escriu **VS+GS+FS** per a mostrar, a banda dels objectes de l'escena, un rectangle texturat centrat a la posició de la llum:



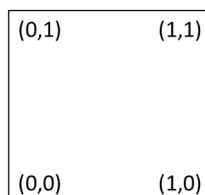
El VS farà les tasques per defecte.

El GS emetrà cada triangle com habitualment. A més a més, si `gl_PrimitiveIDIn` és 0, emetrà un quadrat centrat a la posició de la llum (`uniform vec4 lightPosition`), amb els costats paral·lels als eixos X, Y del sistema de coordenades NDC. El quadrat tindrà costats de mida $2*w$ en NDC, on w és un uniform:

```
uniform float w = 0.3;
```

El **FS** farà les tasques per defecte, tret dels fragments que corresponen al quadrat anterior, pels quals haurà d'aplicar la textura `light.png` amb coordenades de textura en $[0, 1]$ que rebrà del GS. En aquest cas, el color del fragment serà el color de la textura multiplicat pel canal alfa de la pròpia textura. Si la component alfa és inferior a 0.1, el FS descartarà el fragment.

Les coordenades de textura del quadrat han de ser (tal i com es projectarà al viewport):



Podeu fer que el GS emeti coordenades de textura negatives pels triangles de l'escena, i coordenades de textura en $[0,1]$ pel quadrat de la llum, de forma que el FS pugui distingir quin tipus de fragment està processant.

Identificadors obligatoris:

```
show-light.* (minúscules!)  
uniform vec4 lightPosition;  
uniform float w = 0.3;  
uniform sampler2D colorMap;
```

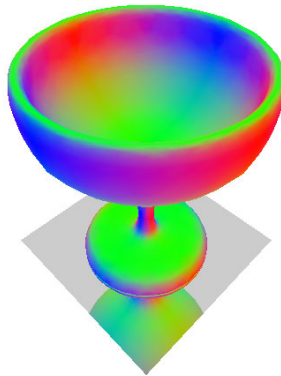
Reflection (reflection.*)

*Copieu aquest plugin a la vostra carpeta `NewViewer_c2dcae3/plugins/` i obriu-lo amb **GLarenaPL**.*

Us proporcionem (amb aquest enunciat) un **plugin reflection** que simula la reflexió de l'escena respecte un mirall que coincideix amb la base de la capsa englobant de l'escena. Suposant que el punt mínim de la capsa és $(\min X, \min Y, \min Z)$, la base està sobre el pla $Y=\min Y$.

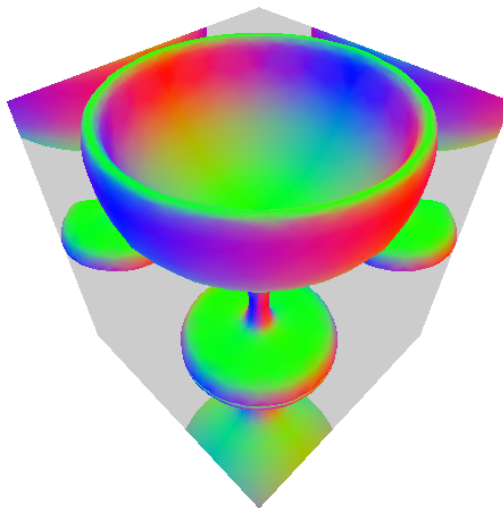
El plugin usa la tècnica de reflexió basada en textures dinàmiques.

Si l'obriu amb GLarenaPL, veureu una imatge com aquesta:



Modifica aquest plugin (useu aquest mateix nom) per tal d'incloure també les reflexions sobre dos miralls més, que coincideixin amb les cares de la capsa englobant de l'escena situades sobre els plans $X=\min X$ i $Z=\min Z$. A cada mirall només cal reflectir l'objecte; ignoreu reflexió dels altres miralls.

Aquí tens el resultat esperat (amb la càmera **Camera** → **Preset 2** del menú):



Identificadors obligatoris:

`reflection.h`, `reflection.cpp`, `reflection.pro`