

Walls2 (walls2.*)

3.5 punts

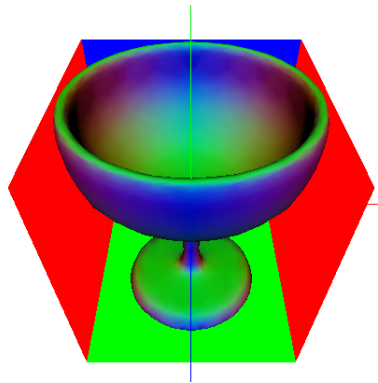
Useu **GLarenaSL** per a desenvolupar aquests shaders (o /assig/grau-g/viewer als PCs dels laboratoris).

Escriu **VS+GS+FS** que dibuixin cada triangle del model de la manera habitual, i a més a més les cares de la capsa englobant de l'escena (boundingBoxMin, boundingBoxMax).

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

El **GS** escriurà cada triangle de l'escena de la manera habitual. El color de cada triangle serà el color per defecte que envia el VS, **tret que la càmera sigui dins la capsa englobant**, ja que en aquest cas el color per defecte caldrà multiplicar-lo per 2.

A més a més, quan **gl_PrimitiveIDIn** sigui 0, el GS haurà de dibuixar 4 de les 6 cares de la capsa englobant de l'escena. El color d'aquestes cares dependrà de l'orientació: vermell per cares perpendiculars a l'eix X, verd per les perpendiculars a l'eix Y, i blau per a les altres dues cares. No dibuixeu ni la cara superior (normal apuntant cap a +Y) ni la cara frontal (normal apuntant cap a +Z):



El **FS** simplement escriurà el color que li arriba del VS.

Per aquest exercici sí teniu un fitxer de test.

Identificadors obligatoris:

```
walls2.* (minúscules!)  
uniform vec3 boundingBoxMin, boundingBoxMax;
```

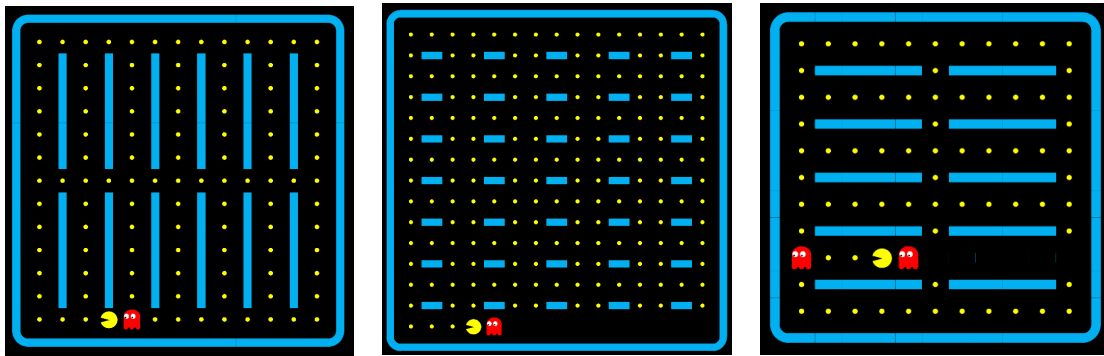
Pacman2 (pacman2.*)

3.5 punts

Useu **GLarenaSL** per a desenvolupar aquests shaders (o /assig/grau-g/viewer als PCs dels laboratoris).

Escriu **VS+FS** per a texturar l'objecte **plane.obj** de forma que mostri una aproximació del que seria una captura de pantalla d'alguna versió plausible del conegut videojoc Pacman.

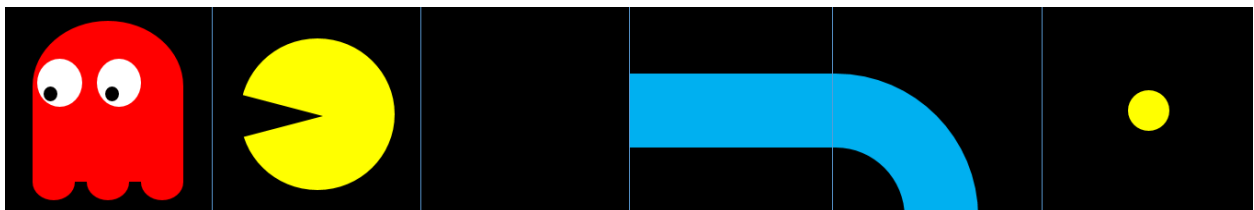
Aquí teniu alguns exemples només orientatius (podeu variar la mida i disposició dels diferents elements):



El **VS** farà les tasques habituals imprescindibles.

El **FS** usará les coordenades de textura, que per l'objecte plane estan dins l'interval $[0,1]$, per a determinar de quin color cal pintar el fragment.

Heu de fer servir la textura **pacman2.png**, amb un escalat i offset adients. Observeu que conté 6 subimatges: fantasma, Pacman, cel·la buida, paret horitzontal, cantonada, i píndola:



Tant els obstacles (parets) del laberint, com les píndoles que menja el Pacman **s'hauran de pintar usant la textura** (i per tant hauran de canviar si es canvia la textura), mirant de reproduir quelcom similar a les versions més conegudes del joc.

Elements que ha de contenir la imatge resultant (la puntuació dels diferents apartats és orientativa):

- Pacman en una posició arbitrària del laberint distinta de l'origen, i amb una mida raonable [**3 punts**]
- Dos fantasmes en posicions arbitràries del laberint [**2 punts**]
- Píndoles; no cal que hi siguin totes, i no patiu si n'hi ha d'inaccessibles [**2 punts**]
- Parets del laberint; quelcom que s'assembli a un laberint, amb parets verticals i horitzontals [**2 punts**]
- Parets exteriors del laberint usant les cantonades arrodonides, orientades correctament [**1 punt**]

Identificadors obligatoris:

pacman2.* (minúscules!)
uniform sampler2D colormap;

Is A Floor? (isafloor.*)

3 punts

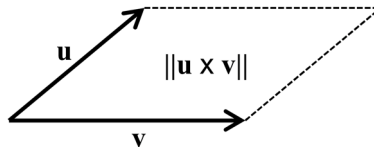
Useu **GLarenaPL** per a desenvolupar aquest plugin.

Escriu un **plugin** que ens permeti visualitzar la mesura en què un model pot fer de terra d'un escenari 3D.

Considerem que una cara serveix com a terra si el **producte escalar** de la seva **normal** unitària per cara i el vector **(0,0,1)** en *world space* dona un valor **superior a 0,7**. El plugin (al mètode onPluginLoad) haurà de calcular la **proporció** λ de la superfície del model que compleix aquesta condició, i escriure aquest valor per la consola (cout), en el format de l'exemple:

TERRA: 0.3423

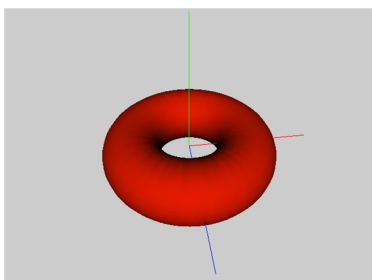
Podem considerar la superfície d'un model com la suma de les àrees de les seves cares. Podeu assumir que els models seran malles de triangles i que només hi haurà un model a l'escena. L'àrea d'un triangle es pot calcular tenint en compte que el **mòdul del producte vectorial** de dos vectors **u** i **v** dona l'àrea del paral·lelogram definit per **u** i **v**, (i per tant el doble de l'àrea del triangle definit per aquests vectors).



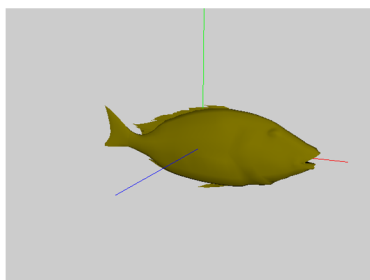
El plugin haurà de passar (en un uniform) la proporció λ als **VS+FS** que s'activaran al **preFrame()**. El **VS** calcularà el color de tots els vèrtexs com a **interpolació lineal** entre vermell (cap triangle és pot considerar terra) i verd (tots ells ho poden ser), usant la proporció λ com a paràmetre de la interpolació. Per a poder visualitzar correctament la forma del model, el **VS modularà** el color amb la component z de la normal en *eye space*.

Si utilitzeu fitxers pels shaders, **utilitzeu paths relatius!**

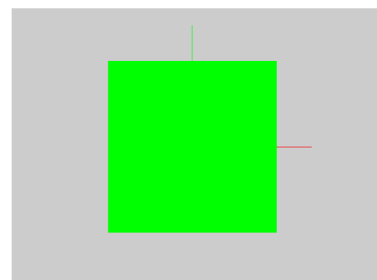
Aquí teniu el resultat esperat per diferents models:



TERRA: 0.104186



TERRA: 0.475672



TERRA: 1

Identificadors obligatoris:

isafloor.pro, isafloor.cpp, isafloor.h (*minúscules!*)