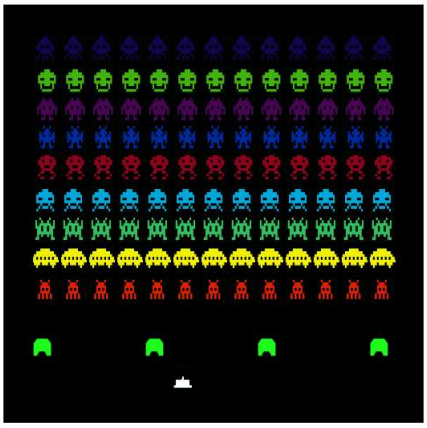
Invaders (invaders.*) Useu: `~/assig/grau-g/Viewer/GLarenaSL`

Escriu **VS+FS** per tal de dibuixar quelcom similar a una pantalla del conegut *Space Invaders*. Aquí teniu la textura invaders.jpg, i un exemple del resultat esperat (amb l'objecte plane):



Observeu que la darrera columna de la textura inclou un canó blanc i un escut verd.

El **VS** farà les tasques imprescindibles.

El **FS** serà l'encarregat de triar el color del fragment, d'acord amb les coordenades de textura que rebrà del VS, i que per l'objecte plane estan dins [0,1].

Per tal d'aconseguir la màxima puntuació, caldrà que:

- Hi hagi un canó blanc a la part inferior (4 punts)
- Hi hagi alguns escuts en una fila per sobre del canó (3 punts)
- Hi hagi com a mínim 6 fileres amb els extraterrestres invasors; cada filera mostrarà un únic tipus d'extraterrestre (3 punts).

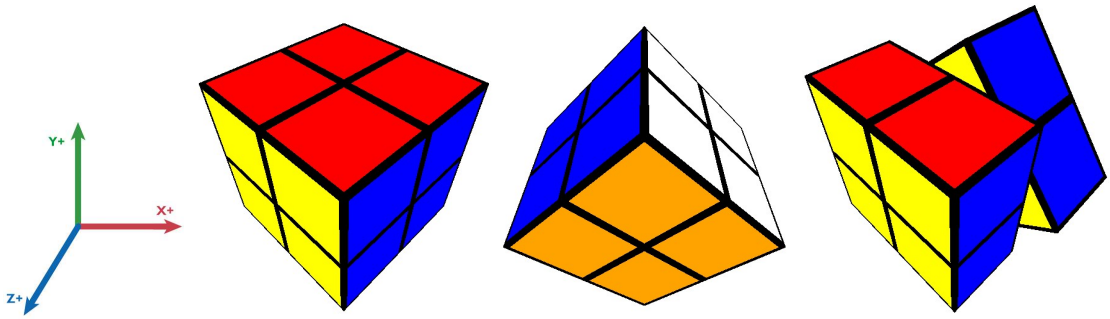
El test és només orientatiu.

Identificadors obligatoris:

`invaders.vert, invaders.frag` (has escrit **invaders** correctament? En minúscules?)
`uniform sampler2D colormap;`

Rubiks (rubiks.*) Useu: `~/assig/grau-g/Viewer/GLarenaSL`

Escriu **VS+GS+FS** per tal de dibuixar un cub de Rubik simplificat:



El **VS** farà les tasques imprescindibles.

El **GS** serà l'encarregat de dibuixar els cubs (un cub per cada execució del GS). El GS sabrà quin dels 8 cubs ha d'emetre pel valor de gl_PrimitiveIDIn. Si gl_PrimitiveIDIn >= 8, el GS no emetrà cap primitiva.

Els cubs tindran longitud d'aresta 2, i estaran centrats en els següents punts (en *object space*):

`(-1,-1,-1), (1,-1,-1), (-1,1,-1), (1,1,-1), (-1,-1, 1), (1,-1, 1), (-1,1,1), (1,1,1)`

El color de les cares del cub serà, segons l'orientació de la seva normal (tot i que no usareu il·luminació):

bottom (y negativa):	orange RGB (1, 0.6, 0);	top (y positiva): red
left (x negativa):	green	right (x positiva): blue
back (z negativa):	white	front (z positiva): yellow

Abans de passar els vèrtexs a *clip space*, el GS aplicarà una rotació de θ=*time* radians al voltant de l'eix Z, però només si gl_PrimitiveIDIn < 4 (és a dir, per 4 dels 8 cubs). Recordeu que la matriu de rotació al voltant de l'eix Z és:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

El **FS** assignarà al fragment el color que li arribi del GS, tret d'un petit marge al voltant de cada cara, que serà de color negre. Per aquest marge, podeu fer que el GS generi coordenades de textura (s,t) per cada vèrtex, entre 0 i 1, i que el FS utilitzi el color de la cara si s, t estan dins [0.05, 0.95]; altrament, el color serà negre. Puntuació orientativa:

- Cubs amb la mida i posició que es demana: 5 punts
- Cares amb els colors que es demanen: 2 punts
- Marge negre a les cares: 2 punts
- Rotació: 1 punt

Identificadors obligatoris:

`rubiks.vert, rubiks.geom, rubiks.frag` (segur que has escrit **rubiks** correctament?)
Tots els uniforms de l'enunciat.

Skyplane (skyplane.*)

Useu `GLarenaPL` de la vostra instal·lació local del visualitzador

Escriu un **plugin** que, donats els shaders **que us proporcionem** (sky.vert, sky.frag, mirror.vert, mirror.frag) permeti visualitzar els objectes com si fossin un mirall que reflecteix una imatge 360º (sky.jpg), la qual també es visualitzarà al fons del viewport usant un rectangle.

De manera més detallada el que es demana és el següent:

- A l'executar el plugin, aquest haurà de llegir, compilar i muntar els shaders següents: *sky.vert*, *sky.frag*, *mirror.vert* i *mirror.frag*. A continuació haurà de llegir la imatge *sky.jpg* crearà la textura amb les configuracions necessàries.

- També un cop per execució del plugin, es crearà un VAO i un VBO per visualitzar un rectangle que ompli el viewport, els vèrtexs del qual estaran en *ClipSpace* el més a prop possible del *far plane*.

- A cada frame (paintGL), el plugin farà aquestes tasques:

    1. Pintarà el rectangle texturat, usant els shaders *sky.vert* i *sky.frag*. Recordeu configurar les textures i enviar els uniforms necessaris.

    2. Pintarà l'escena usant els shaders *mirror.vert* i *mirror.frag*. Recordeu enviar els uniforms necessaris.

Per a aquest plugin no ni ha test.

Identificadors obligatoris:

`skyplane.h, skyplane.cpp, skyplane.pro` (únicament aquests tres fitxers, i en minúscules)