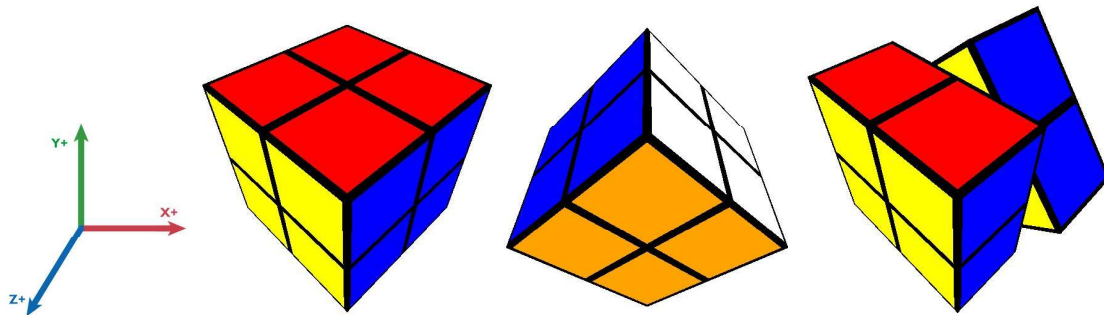


Rubiks (rubiks.*) Useeu: ~/assig/grau-g/Viewer/GLarenaSL

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 $\theta = \text{time}$ 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.