EJERCICIOS (CLASE 08)

1. Usa `glMultMatrixf()` y `glMultTransposeMatrixf()` para crear unas funciones que reproduzcan

   - `glTranslatef()`
   - `glScalef()`
   - `gluLookAt()`

   Para este último, usa `numpy.linalg.norm()` y `numpy.cross()` para definir la base $\{\vec{u}, \vec{v}, \vec{n}\}$. Recuerda también que la matriz asociada a `gluLookAt()` es la composición de una traslación con una rotación.

2. En principio, hemos escrito cualquier rotación $R_{\vec{u}}(\theta)$ como producto de cinco rotaciones

   $$R_{\vec{u}}(\theta) = R_{e_2}(\varphi)^{-1} R_{e_3}(\psi)^{-1} R_{e_1}(\theta) R_{e_3}(\psi) R_{e_2}(\varphi)$$

   ¿Qué ocurre cuando intentamos escribir la rotación $R_{e_2}(\theta)$ como producto de estas cinco rotaciones?

3. Implementa una versión de `glRotatef()` usando las rotaciones de Euler (atento al ejercicio anterior). No olvides normalizar el vector $\vec{u}$, eje de la rotación.

4. Intenta reproducir los argumentos que hemos hecho para describir una rotación $R_{\vec{u}}(\theta)$ como producto de cinco rotaciones de Euler de la forma

   $$R_{e_3}(\varphi)^{-1} R_{e_2}(\psi)^{-1} R_{e_1}(\theta) R_{e_2}(\psi) R_{e_3}(\varphi).$$

   Observa que en este caso (y al contrario de como lo hemos hecho en clase) primero rotamos sobre el eje Z, y luego sobre el eje Y.