

Environment map e riflessione/rifrazione

Texture che rappresenta un'ambientazione lontana, per esempio l'orizzonte (*skybox*). Con *cube mapping*, si disegna la texture su un cubo che contiene l'intera scena. Il cube mapping si può usare anche per i riflessi, ma solo se la texture rappresenta oggetti lontani perché il texel selezionato dipende solo dalla direzione di riflessione (e non dalla posizione), a meno che non sia generata dinamicamente.

- due passi di rendering, prima lo skybox (con il suo shader e `gl_deptMask(false)`) e poi si sovrascrive con la scena;
- disegniamo un cubo che logicamente racchiude tutta la scena (in realtà basta che sia visibile);
- per accedere alla cube map usiamo la posizione dei vertici del cubo in `view* space` interpolata e normalizzata (ovvero la direzione dal centro del mondo al frammento);
- `gl_Position = uProjectionMatrix * vec4((uViewMatrix * vec4(aPosition, 0.0)).xyz, 1.)`, non consideriamo la traslazione della view matrix.

Per la riflessione si usa il vettore riflesso portato in world space e normalizzato per accedere alla cube map, analogamente per la rifrazione. Per calcolare il vettore di rifrazione (senza usare `refract`) si può usare la formula di Rodriguez sull'asse $N \times V$ per ruotare $-N$ per l'angolo di rifrazione (non serve calcolarlo, si lavora solo su seni e coseni calcolati con prodotto scalare e $\sin^2 + \cos^2 = 1$).

La rifrazione andrebbe calcolata anche in uscita ma non si può con la pipeline. Per combinare riflessione e rifrazione (senza altro colore/illuminazione):

```
gl_FragColor = schlick * reflected + (1. - schlick) * refracted;
```

Per generare dinamicamente la cube map, si fanno 6 rendering (in realtà 12, anche skybox) con centro nell'oggetto per cui useremo la cube map (che non va incluso nel rendering), ciascuna con direzione di vista in una delle 6 direzioni cardinali, FOV di $\pi/2$ e aspect ratio 1.