

31/10/2016

FACULTATEA
DE
AUTOMATICA SI
CALCULATOARE

SISTEME DE PRELUCRARE
GRAFICA

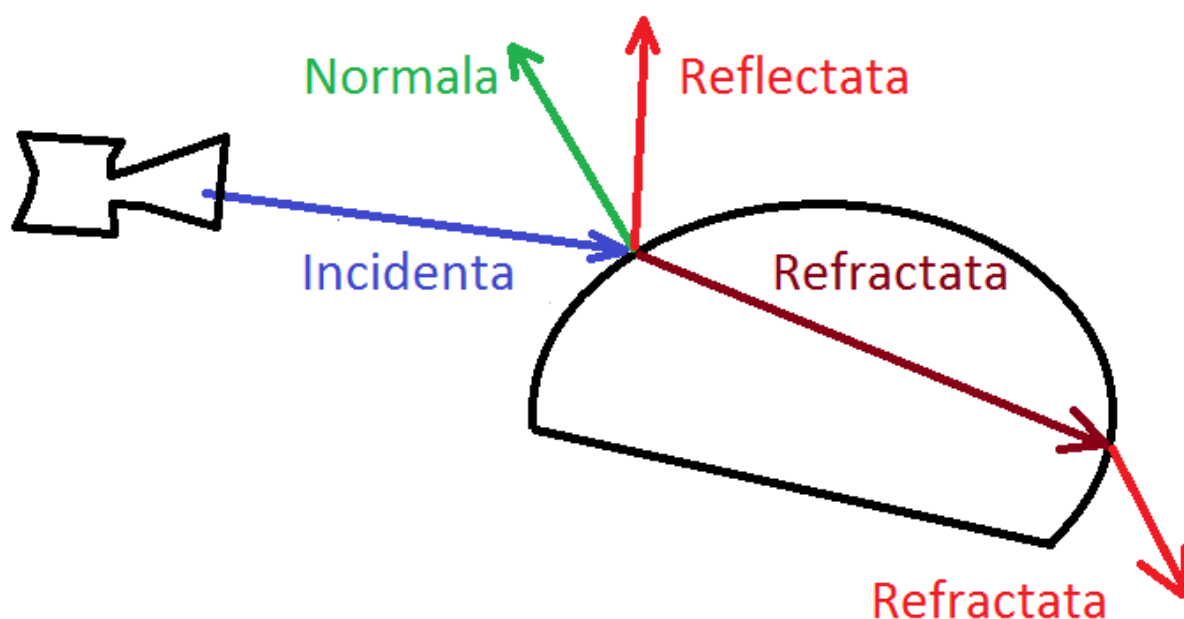


Laborator 4

Maparea mediului inconjurator

Introducere

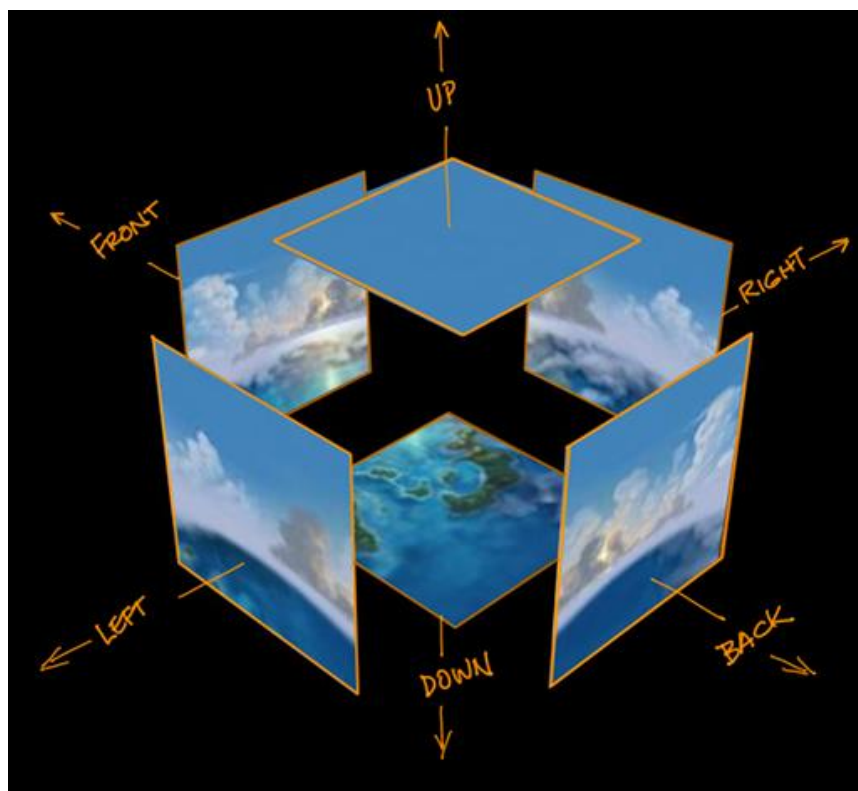
Maparea mediului inconjurator reprezinta o metoda prin care putem reda pe suprafata unui obiect obiectele din mediul inconjurator care se reflecta sau refracta pe suprafata sa. Fie ca facem acest lucru pentru reflexii, refractii sau alte efecte mai complicate ideea este aceeaasi: de a putea simula intr-un mod eficient mediul inconjurator.



Maparea mediului inconjurator este o tehnica dependenta de pozitia observatorului, raza reflectata/refractata fiind calculata in functie de pozitia observatorului si normala la suprafata. In imaginea de mai sus se poate observa cum o raza ce pleaca din camera se poate descompune in doua traiectorii distincte: una reflectata si una refractata. In realitate procesul este mai complicat dar pentru o iluzie vizuala acest model este suficient.

Ce este un cubemap

Un cubemap este un tip special de textura care este formata din 6 imagini: pozitiv x, negativ x, pozitiv y, negativ y, pozitiv z, negativ z. Fiecare imagine a unei texturi cube map reprezinta una dintre cele 6 imagini panoramice ale mediului care inconjoara obiectele texturate: impreuna ele formeaza imaginea mediului inconjurator, asa cum poate fi observat in imaginea urmatoare. Deci, un cubemap poate fi perceput ca un set de elemente de tip (directie, culoare). Un cubemap poate fi citit doar cu coordonate de texturare tridimensionale, pentru ca interogam culoarea din harta pentru directia (coordonata de texturare) introdusa. Mai mult cum coordonatele de texturare sunt de fapt directii, ele pot fi si negative.



Formatul in care veti gasi de obicei acest tip de textura este ca 6 imagini diferite sau ca o imagine cu urmatoarea forma.



Un cubemap este creat la fel ca orice alta textura, cu comanda:

```
glGenTextures(1, &gl_texture_object);
```

Cubemapul este legat la banda grafica la un nou punct de legare per unitatea de texturare curent legata, numit GL_TEXTURE_CUBE_MAP:

```
glBindTexture(GL_TEXTURE_CUBE_MAP, gl_texture_object);
```

Din punct de vedere al filtrarii procesul este identic cu cel al unei texturi bidimensionale, doar ca se poate folosi si:

```
glEnable(GL_TEXTURE_CUBE_MAP_SEAMLESS);
```

care imbunatateste efectele vizuale la filtrarea de la marginea dintre fete.

Pentru a introduce date in aceasta textura, se introduc pe rand toate cele 6 imagini, fiecare pe o pozitie specifica:

```
glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_X, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_posx);
glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_Y, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_posy);
glTexImage2D(GL_TEXTURE_CUBE_MAP_POSITIVE_Z, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_posz);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_X, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_negx);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_Y, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_negy);
glTexImage2D(GL_TEXTURE_CUBE_MAP_NEGATIVE_Z, 0, GL_RGB, width, height, 0, GL_RGB,
GL_UNSIGNED_BYTE, data_negz);
```

Shadere

Pentru a fi utilizat in shader, un cubemap trebuie legat la banda grafica, proces identic cu cel pentru o textura 2D normala. In schimb, in shader se foloseste un sampler diferit, numit **samplerCube** care functioneaza doar cu coordonate de texturare tridimensionale.

Pentru a implementa reflexia si refractia se pot folosi functiile de glsl:

reflect(RazaIncidentă, Normala)

refract(RazaIncidentă, Normala, indexMediuCurent/indexMediuNou)

ce ne ofera directia razei reflectate sau refractate. Dar cum cubemapul reprezinta de fapt o harta de culoare interogata prin directie, putem folosi razele reflectate/refractate din fragment ca si coordonate de texturare pentru a afla culoarea mediului din directia razelor.

Rezultatul final pentru o reflexie poate fi calculat cu:

```
vec3 culoarefragment = texture(cubemap, reflect(RazaIncidentă, Normala));
```

Observatii:

- pentru refractia unei raze din aer (index de refractie 1.0) in apa (index de refractie 1.33) apelul de functie ar fi `refract(I, N, 1.0/1.33);`
- raza incidenta vine de la camera la suprafata, nu invers!

Rezultate(reflexie, apoi refractie)

