

Ceata

Pana acum s-au folosit diverse tehnici pentru a nu procesa mai multe date deca trebuie sa afisam:

- frustrumul de vizualizare
- Generarea terenului numai pana la o anumita distanta fata de camera.

Totusi, cand ne deplasam in spatiu si anumite obiecte trec in zona vizibila, afisarea lor poate parea brusca. La fel si generarea unei noi fasii de teren daca acesta nu depaseste skyboxul astfel incat generarea sa fie ascunsa de obiectul skybox. De aceea vom folosi un efect de ceata (fog).

In fisierul de setari pentru scena se va preciza culoarea cetii (prin componentele r, g, b). Ideal ar fi ca aceasta culoare sa coincida cu culoarea din partea de jos a skyboxului, pentru a avea un efect vizual placut.

Aplicarea cetii presupune urmatoarele:

- cunoasterea coordonatelor camerei
- Existenta unei zone de claritate in care vedem in scena culorile proprii obiectelor asa cum le-am vazut si pana acum. Vom nota raza zonei de claritate cu r .
- Existenta unei zone de tranzitie in care culoarea obiectului e combinata cu culoarea cetii. Vom nota cu R distanta (raza) la care se termina zona de tranzitie si incepe urmatoarea zona (cea de ceata)
- Zona de ceata (fog) in care obiectele (sau partile de obiecte) au culoarea cetii.

In zona de tranzitie se va folosi un numar real alpha intre 0 si 1 (0 corespunzand distantelor mai mici decat r si 1 distantelor mai mari decat R), care va creste Iniar intre r si R in functie de distanta (cand distanta e intre r si R , alpha este in intervalul $[0,1]$ avand o valoare proportionala cu distanta). Astfel culoarea in zona de tranzitie va fi:

$$\text{culoare} = \alpha * \text{culoare_fog} + (1.0 - \alpha) * \text{culoare_obiect}$$

Pentru un calcul usor in shader se va folosi functia [clamp](#) .

Pasi de urmat:

1. Se definesc sceneManager.xml raza mica, raza mare si culoarea cetii. Definirea se va face direct in elementul sceneManager (acestea nu sunt proprietati ale obiectelor ci ale scenei). Culoarea cetii se preia din textura pentru skybox (culoarea din fateta y-negativ). Initial (pentru testare si debug) se poate pune o culoarea mai usor de distins:alb, rosu etc. Atentie, componentele culorii sunt in intervalul [0,1] si nu [0,255].
2. Deoarece lucram cu culori si nu cu pozitii, vom defini in fragment shader variabile de tip uniform pentru raza mica (float), raza mare(float) si culoarea cetii(vector cu 3 componente).
3. Preluam locatiile in clasa Shader, si apoi transmitem cele 3 informatii (cautati functiile de tip glUniform potrivite: <https://www.khronos.org/registry/OpenGL-Refpages/es2.0/>). Aceste informatii ar trebui sa fie valabile pentru toate obiectele.
4. Conform teoriei, pentru a calcula distanta de la camera la fragment, avem nevoie de pozitia camerei si pozitia in scena corespunzatoare fiecarui fragment generat.
 - a) Pentru pozitia corespunzatoare fragmentului, va fi nevoie sa ne facem un varying cu pozitia (care va fi calculata automat prin interpolare). Aceasta se va calcula in vertexShader, pornind de la pozitia **in scena** a vertecilor (fiind vorba de pozitia in scena si nu in local space, inseamna ca ne trebuie pozitia dupa inmultirea cu matricea *model* (world), dar fara inmultirea cu *view* si *perspective* – deci e posibil sa aveti nevoie sa transmiteti separat matricea *model*).
 - b) Cream un uniform pentru pozitia camerei (un vector de 3 componente), pe care il transmitem in fragment Shader.
 - c) Calculam distanta dintre pozitia camerei si pozitia fragmentului cu functia de shader: **distance** (care primeste ca parametri punctele intre care trebuie calculata distanta)
5. Consideram distanta notata cu d . Trebuie sa calculam alpha (folosind functia de shader **clamp**) astfel incat:
 - a) daca $d \leq r$ alpha=0
 - b) daca $d \geq R$ alpha=1
 - c) daca $r < d < R$ atunci $0 < \alpha < 1$ astfel incat $\alpha = (d-r)/(R-r)$ ($R-r$ e dimensiunea zonei de tranzitie de la claritate la ceata, iar $d-r$ e distanta de la **pozitia** fragmentului din zona de tranzitie la **inceputul** zonei de trecere).Gasiti o modalitate de a-l calcula pe alpha fara a folosi if; mai precis, folositi **clamp**.
6. Aplicati formula precizata in teorie. Considerand ca avem in culoare_obiect culoarea obiectului (eventual cu alte efecte aplicate: lumini etc), culoarea cu ceata aplicata, va fi:
$$\text{culoare} = \alpha * \text{culoare_fog} + (1.0 - \alpha) * \text{culoare_obiect}$$

Aceasta va fi setata in **gl_FragColor**.