

Proiect la disciplina Prelucrare Grafica

Mudura Ana-Andreea
grupa 30231

January 16, 2023

Contents

| | | |
|----------|---|----------|
| 1 | Prezentarea temei | 1 |
| 2 | Scenariul | 1 |
| 2.1 | Descrierea scenei si a obiectelor | 1 |
| 2.2 | Functionalitati | 2 |
| 3 | Detalii de implementare | 2 |
| 3.1 | Functii de algoritmi | 2 |
| 3.1.1 | Solutii posibile | 2 |
| 3.2 | Structuri de date | 3 |
| 3.3 | Ierarhia de clase | 3 |
| 4 | Modelul grafic | 3 |
| 5 | Manual de utilizare | 6 |
| 6 | Dezvoltari ulterioare | 6 |

1 Prezentarea temei

Proiectul consta in realizarea unei aplicatii controlate de la consola ce consta in exemplificarea unei prezentări fotorealiste a unor scene de obiecte 3D utilizând librăriile:

- OpenGL
- GLM
- GL
- GLM

Pe langa asta, utilizatorul trebuie sa aiba posibilitatea de a naviga prin scena folosind tastatura sau mouse-ul.

2 Scenariul

2.1 Descrierea scenei si a obiectelor

Scenariul creat este o scena de natura, realizata si texturata in aplicatia Blender care are aspectul unei forme de relief deal. Pentru a exemplifica si functionalitatea unui SkyBox, scena de natura pluteste astfel exemplificandu-se scenariul SkyBoxului prezentat in laboratoare.

In mijlocul naturii se afla o scena, la care instrumentistii sunt animale. In jurul scenei texturata

holografic se afla copaci si tufe, iar pe deasupra, pe cer zboara un avion. Pentru atmosfera muzicala in scena se mai afla si un discoball care are reflexia skyboxului.

Pentru scenariul de noapte una dintre cele 2 surse de lumina reprezinta un reflector ce lumineaza deasupra scenei cu o lumina mov pentru a da un efect de petrecere.

Pentru a exemplifica efectul de transparenta in scena mai exista baloane ce sunt lipite de scena de concert, iar la baza acestora se afla o sticla.

2.2 Functionalitati

Pentru inceput functionalitatile proiectului incep prin navigarea prin scena. Aceasta se poate realiza prin miscarea mouseului, astfel camera se va misca dupa cursor sau prin apasarea tastelor W,A,S,D cu care se poate naviga fata-spate-stanga-dreapta.

Scena se poate vizualiza in diferite moduri, cum ar fi wireframe,solid,point, poligon si smooth prin apasarea tastelor 1,2,3,4 si respectiv 5.

Pentru rotirea discoballului se apasa pe tasta P, iar pentru activarea miscarii avionului se apasa tasta K.

In proiect exista doua surse de lumina: cea realizata dupa modelul de Iluminare Phong, iar cea de a doua este o lumina punctiforma. Prima sursa de lumina este alba, ea se activeaza de pe tastele J si L (rotindu-se in jurul unei axe), iar la rotirea ei umbrele obiectelor se misca in functie de pozitia cubului.

A doua sursa de lumina punctiforma este pozitionata in locul reflectorului scenei tocmai pentru a simula functionalitatea unei lumini de scena. Pentru a activa lumina scenei se apasa pe tasta B, iar pentru a o dezactiva se apasa tasta N.

Scena mai detine si efectul de ceaata care acesta poate sa fie activat de pe tasta Z si dezactivat de pe tasta X.

Pentru pornirea animatiei de prezentare care va porni navigarea in scena fara a folosi mouseul sau tastatura se va apasa pe tasta F.

3 Detalii de implementare

3.1 Functii de algoritmi

3.1.1 Solutii posibile

Pentru cele doua surse de lumina s-au folosit doi algoritmi diferiti:

- lumina ambientala,speculara,difuza a fost calculata in fragment shader, ele fiind o combinatie de 3 vectori de 3 elemente (ambientala fiind un produs intre culoarea trimisa si un numar real de ii reprezinta intensitatea,difuza un maxim intre 0 si produsul scalar dintre directia luminii si normal eye, iar speculara un produs intre coeficientul de intensitate calculat,culoarea trimisa si directia luminii)
- lumina punctiforma pe langa cele mentionate la punctul anterior mai contine 3 constante: una constanta, una quadratica si una liniara plus un coeficient de atenuare care este calculat folosind cele trei constante mentionate anterior

Pentru rotirea obiectelor, cum ar fi miscarea avionului deasupra scenei sau rotirea celor 3 globuri deasupra scenei de concert s-au folosit operatii de scalare, translatie si rotire pe axa X, respectiv Y.

Cele 3 obiecte reflectorizante au fost texturate dintr-un shader separat care aplica culoarea reflectata de la skybox in loc de culoarea trimisa printr-o variabila uniform

Efectul de ceaata a fost realizat din fragment shader prin functia computeFog care returneaza o variabila flotanta calculata printr-o formula ce face produsul la o densitate dorita si lungimea vectorului ce contine pozitia fragmentului in eye space. Numarul real returnat de functie este folosit in a calcula noua culoare cu ajutorul unui vector de culoare pentru ceaata. Pentru a controla activarea/dezactivarea cetii se trimite o variabila uniform shaderului.

Efectul de transparenta a avionului, a sticlei si a baloanelor s-a folosit modificand alpha-ul de opacitate atribuit variabilei fColor din fragment shader si folosind functia glEnable si GLBlendFunc ce foloseste alpha-ul trimis pentru a crea efectul de transparenta.

COD

```
glEnable(GL_BLEND);
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
baloane.Draw(myCustomShader);
sticla.Draw(myCustomShader);
glDisable(GL_BLEND);
Efectul de umbre a fost realizat tot din shaderele obiectului prin functia de compute shadow
```

COD

```
float computeShadow()
vec3 normalizedCoords = fragmentPosEyeSpace.xyz / fragmentPosEyeSpace.w;
normalizedCoords = normalizedCoords * 0.5 + 0.5;
float closestDepth = texture(shadowMap, normalizedCoords.xy).r;
float currentDepth = normalizedCoords.z;
float bias = max(0.05f*(1.0f - dot(fNormal, lightDir)), 0.005f);
float shadow = currentDepth - bias < closestDepth ? 1.0 : 0.0;
return shadow;
```

Pentru rasterizarea scenei cu umbre trebuie initial creata o harta, iar in functie de adancimea fragmentului ce trebuie desenat si adancimea hartii acesta se va afla in umbra sau nu (valori de 0 sau 1)(ambele trebuie sa fie in sistemul de referinta a luminii).

Pentru animatia de prezentare am folosit 2 fisiere text care contin coordonatele vectorului de pozitie si de front direction (printate din consola la o rulare). Dupa popularea fisierului acesta in bucla de iteratie din functia principala am actualizat cei doi vectori cu valorile citite din cele 2 fisiere

COD

```
std::ifstream myfile;
std::ifstream myfile2;
std::string vector;
myfile.open("MiscariCamera.txt");
myfile2.open("MiscariDirectie.txt");
...
if (enable_animation)myfile >> myCamera.cameraPosition.x;myfile >> myCamera.cameraPosition.y;myfile >> myCamera.cameraPosition.z;
myfile2 << myCamera.cameraFrontDirection.x;
myfile2 << myCamera.cameraFrontDirection.y;
myfile2 << myCamera.cameraFrontDirection.z;
```

3.2 Structuri de date

In proiect s-au folosit urmatoarele structuri de date

- pentru fisiere std::ifstream
- pentru vectori vec3 vec4 si matrici mat3 mat4 din biblioteca glm
- pentru camera structura de date Camera ce continea vectorii de pozitie,UP,Front direction etc.
- modele,shader am folosit Model3d si Shader din namespace-ul gps
- pentru comunicarea dintre aplicatie si shaderele obiectelor au folosite variabile uniform, respectiv pentru comunicarea intre shader variabile de tip IN/OUT
-

3.3 Ierarhia de clase

In figura 1 se poate vizualiza ierarhia de clase a proiectului

4 Modelul grafic

In figura 2 se poate observa scena realizata cu efectul de ceata activat si lumina punctiforma de culoare roz pornita.

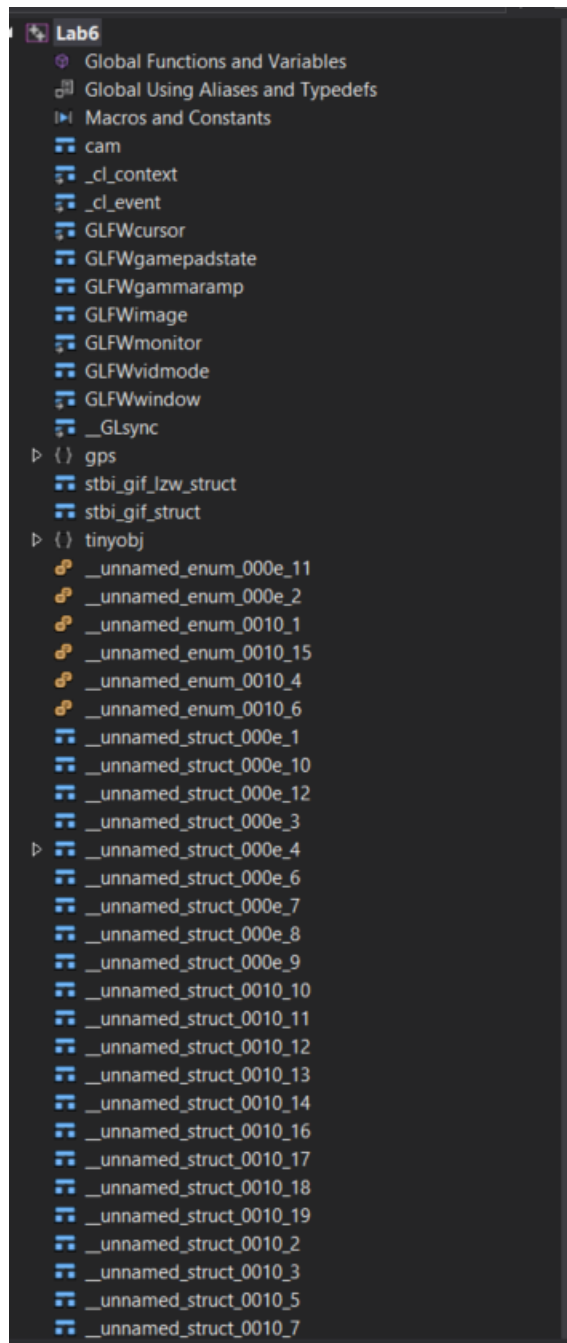


Figure 1: Diagrama de clase a proiectului

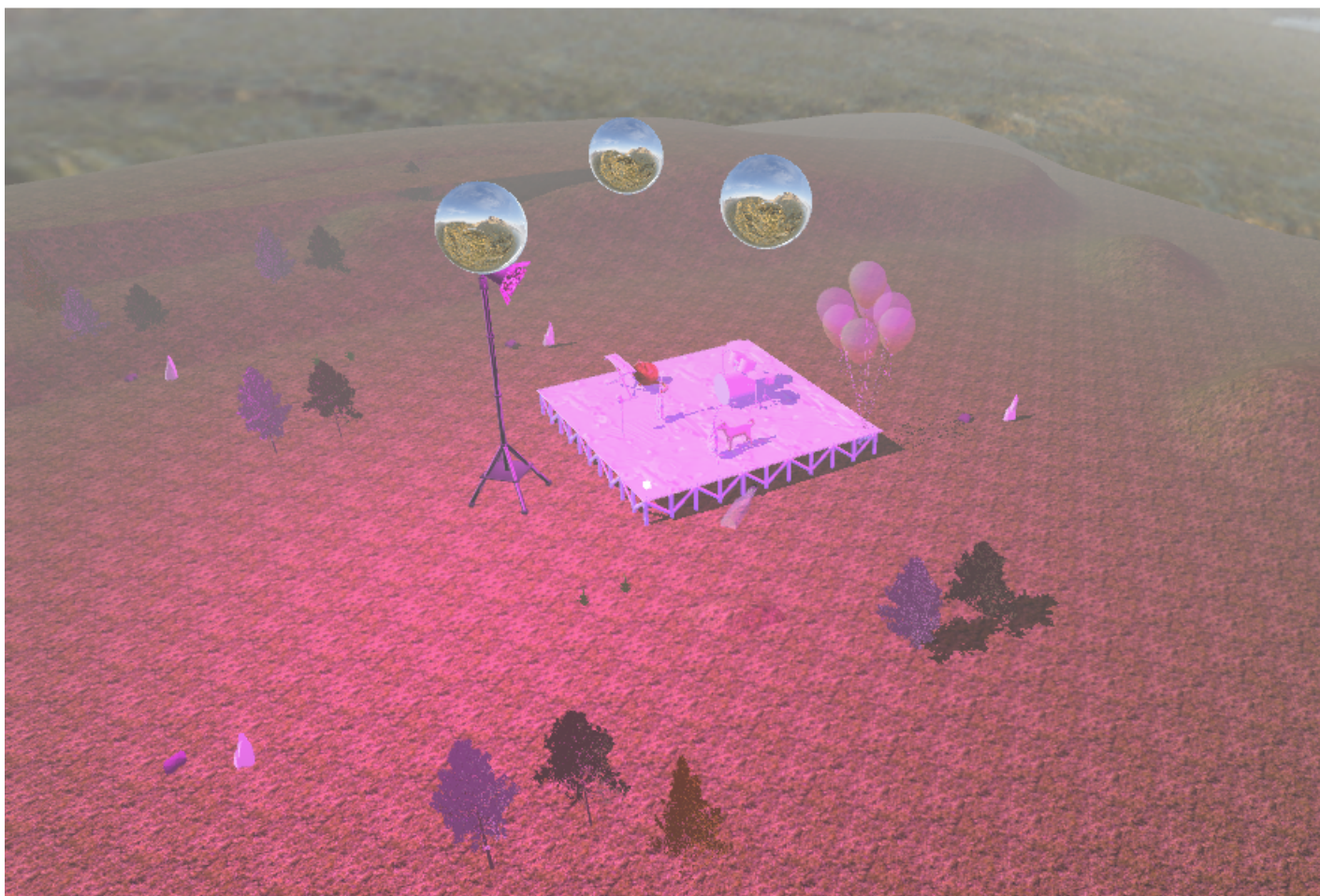


Figure 2: Scena vazuta in ansamblu

5 Manual de utilizare

Proiectul a fost realizat in Visual Studio 2019. Dupa cum am mentionat mai sus, dupa rularea aplicatiei in modul Release:

- W,A,S,D pentru miscare dreapta,stanga,fata,spate
- Q,E pentru rotatii
- 1,2,3,4,5 pentru vizualizarea scenei in diferite moduri
- Z,X pentru activare,dezactivare ceata
- B,N activare,dezactivare lumina punctiforma
- J,L pentru rotirea luminii directionale
- K,p pentru rotirea avionului, mingilor reflectorizante
- C,V pentru activarea,dezactivarea animatiei de prezentare

6 Dezvoltari ulterioare

Scena mai poate fi imbogatita cu alte obiecte sau exstinsa cu mai multe scenarii, renuntatnd la aspectul simplist.

Proiectului ii se mai pot adauga mai multe tipuri de lumini, cum ar fi spotlight sau lanterna.

Obiectele pot sa fie continua animatii mai complexe, cum ar fi animarea unei componente din compozitia sa.

Pe langa cele mentionate mai sus, proiectul pe viitor poate sa contine efecte de ploaie, vant sau ninsoare ce contin un grad de complexitate mai ridicat.

References

Tutoriale: [Learn OpenGL](#)