

Proiect Prelucrare Grafica

Student: Szakacs Emma-Evelin
Grupa: 30231

1. Cuprins

1. Cuprins	2
2. Prezentarea temei	2
3. Scenariul	2
3.1. Descrierea scenei si a obiectelor	2
3.2. Fuctionalitati	3
4. Detalii de implementare	3
4.1. Functii si algoritmi	3
4.1.1. Solutii posibile	3
4.1.2. Motivatia abordarii alese	4
4.2. Modelul grafic	4
4.3. Structuri de date	4
4.4. Ierarhia de clase	5
5. Manual de utilizare	5
6. Concluzii si dezvoltari ulterioare	7
7. Referinte	8

2. Prezentarea temei

Acest proiect a presupus realizarea unei scene de obiecte 3D intr-un mod cat mai realist prin utilizarea librariilor OpenGL, GLFW si GLM. Obiectele 3D au fost modelate in aplicatia Blender, iar apoi transportate ca si fisiere .obj in fisierul proiectului, cu texturile lor aferente. Utilizatorul se poate misca in scena cu ajutorul mouse-ului si a tastaturii si poate activa cateva animatii prin apasarea unor taste.

3. Scenariul

3.1. Descrierea scenei si a obiectelor

Pentru acest proiect am ales sa modelez o scena pe plaja marii. In cadrul scenei se afla o casa pe care am modelat-o de la 0 in Blender si cu cateva obiecte in interior, cum ar fi canapeaua, televizorul si masa. In exterior sunt cativa copaci, o piscina, o masina, niste sezlonguri si doua felinare.

3.2. Fuctionalitati

Am ales sa modelez o scena obscura, folosindu-ma de lumini si ceata. De asemenea, am implementat si reflexia catorva obiecte fata de skybox (mediul incojurator). Printre aceste obiecte se numara geamurile, masina si piscina. Ca si animatie am o usa care se poate inchide sau deschide la apasarea unor taste. De asemenea, tot prin apasarea unor taste pot trece de la modul noapte la modul zi prin schimbarea luminilor. Efectul geamurilor l-am creat prin intermediul transparentei si a reflexiei. Sursele de lumina din scena sunt in numar de 9: lumina principala(directionala), doua lumini apartin felinarelor din jurul piscinei, doua lumini colorate vin din interiorul casei, iar 4 lumini sunt pozitionate in fata farurilor masinii. Umbrele din scena au fost implementate folosind tehnica de mapare a umbrelor prin harta adancimilor. Detaliile implementarii vor fi discutate ulterior.

4. Detalii de implementare

Pentru implementarea proiectului am folosit resursele din laboratorul de Prelucrare Grafica, iar pentru detalierea mai amanuntita a unor notiuni neclare m-am folosit de alte resurse mentionate in laborator.

4.1. Functii si algoritmi

Algoritmii implementati in acest proiect sunt: generarea cetei, maparea umbelor, aplicarea transparentei pentru cateva obiecte si generarea reflexiei si refractiei.

4.1.1. Solutii posibile

Pentru generarea cetei, am ales sa implementez ceata exponentiala patratica pentru a avea un rezultat cat mai realist. Efectul de ceata se obtine prin scaderea intensitatea luminii in functie de distanta la care se afla obiectul respectiv fata de "ochii" camerei. Culoarea cetii se va amesteca cu cea a fragmentului, iar asta va da o nota realista scenei. Alte optiuni ar fi fost ceata liniara si ceata exponentiala liniara.

Pentru a face generarea umbrelor obiectelor aflate in scena, relativ la lumina principala, am folosit metoda hartii umbrelor. Acesta metoda se bazeaza pe texturi de adancime. In prima parte a algoritmului se genereaza harta de adancime a scenei din perspectiva luminii care este salvata ca o textura. In a doua parte a algoritmului textura creeata se ataseaza la shader-ul care calculeaza culorile si se rasterizeaza scena impreuna cu umbrele generate. O abordare mai realista ar fi fost sa folosesc si umbre date de restul lumilor din scena (cele punctiforme). Aceasta metoda este cunoscuta ca mapare omnidirectionala a umbrelor.

Pentru prezentarea scenei de obiecte 3D am ales cateva pozitii din scena de unde sa inceapa miscarea camerei. Acesta parte am implementat-o in functia render scene deoarece este singura functie care se repeta in bucla.

Pentru aplicarea transparenței pe unele obiecte, cum ar fi apa piscinei și geamurile, am folosit tehnica de blending din OpenGL, apelând funcția `glEnable(GL_BLEND)`. Printr-o variabilă „alpha” am trimis shaderului gradul de transparență a obiectului. [2]

Reflexia și refracția[2] le-am calculat în fragment shaderul în care calculez culorile obiectelor, relativ la skybox. Funcțiile `reflect()` și `refract()` sunt funcții deja definite în limbajul shaderelor, iar eu a trebuit doar să calculez normala direcției de vizualizare și norma „ochilor” camerei. Cele două efecte se puteau calcula și prin metode matematice.

4.1.2. Motivatia abordarii alese

Ceata am ales să o implementez exponențială patratică deoarece este cel mai realist mod de implementare a cetei. Pentru umbre am ales maparea hărții umbrelor deoarece este metoda pe care am înțeles-o mai bine la laborator. Pentru celelalte funcționalități am ales abordările mai simple, care nu au necesitat prea multă matematică, ci mai mult înțelegerea bibliotecilor și a funcțiilor din OpenGL.

4.2. Modelul grafic

Modelul grafic al acestui proiect a fost realizat folosind modelul de iluminare Phong, având atât lumina ambientală se și difuza, cât și lumina speculară. În procesul de rasterizare am folosit efectele de umbră, ceață, transparență, reflexie și refracție. Animația usii a fost realizată printr-o translație în origine, o rotație și o translație înapoi în punctul inițial.

4.3. Structuri de date

Structurile de date pe care le-am folosit în acest proiect sunt structurile specifice OpenGL: VAO, VBO, EBO și FBO.

VAO (Vertex Array Object) este un obiect care conține unul sau mai multe VBO-uri și este creat pentru a reține informații necesare rasterizării obiectelor.

VBO (Vertex Buffer Object) este un buffer de memorie care descrie coordonatele sau culorile varfurilor.

EBO (Element Buffer Array) stochează indicii pentru evitarea duplicatelor. Indicii vor fi utilizați de OpenGL pentru a decide ce varfuri să aleaga atunci când formează obiectele.

FBO (Frame Buffer Object) este o colecție de texture sau alte tipuri atasate.

Pe lângă structurile de date specifice OpenGL, am folosit și structuri de date din biblioteca glm, cum ar fi `vec3`, `vec4`, `mat3`, `mat4` etc.

4.4. Ierarhia de clase

Structura proiectului se poate observa in figura de mai jos:

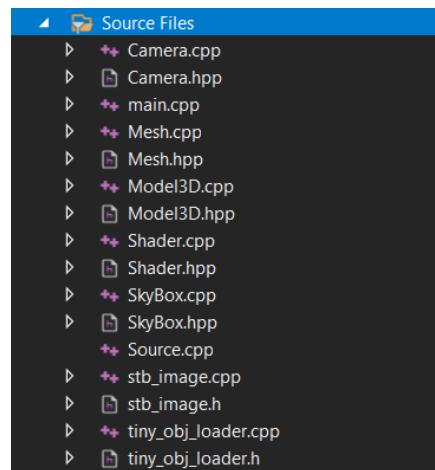


Figure 1 Ierarhia de clase

Clasa “Camera.cpp”, asa cum ii spune si numele defineste parametrii camerei si miscarea acesteia, avand un fisier headere atasat. Clasa “Model3D” este pentru prelucrarea obiectelor 3D din scena, avand o functii de aplicarea a texturii pe obiect, in functie de fisierul .mtl si functii de incarcare a obiectului in scena. Acesta clasa are si ea un fisier header atasat. Clasa “tiny_obj_loader.cpp” si fisierul header atasat ei “header_obj_loader.h” sunt definite in biblioteca OpenGL si sunt folosite tot in modelarea obiectelor 3D. Clasa “SkyBox.cpp” defineste pozitiile varfurilor texturii care va reprezenta “cerul” scenei. Clasa shader defineste functionalitatea shaderelor acestui proiect. Shadererele atasate proiectului se pot observa in figura de mai jos.

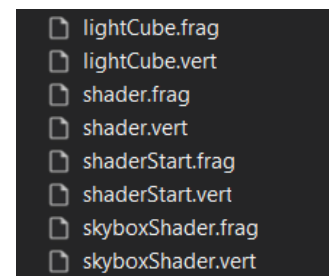


Figure 2 Shadere

Cea mai importanta clasa din proiect, fisierul “main.cpp” este cea care da functionalitate proiectului cu ajutorul claselor mentionate anterior.

5. Manual de utilizare

Pentru inceput, proiectul trebuie deschis intr-un editor de cod, cum ar fi vVisual Studio. Pentru rulare se poate alege modul Debug sau Realise. Dupa ce utilizatorul apasa butonul de rulare, se va deschide o fereastră, iar dupa cateva secunde se va putea vizualiza scena. Tastele au urmatoarele functionalitati:

- Tasta “T” – turul vizual al scenei de obiecte, prezentand cateva cadre.
- Tasta “Y” – oprirea turului vizual

- Tasta "4" – activeaza modul de zi, setand culoarea luminii la (1.0f,1.0f,1.0f)
- Tasta "5" – activeaza modul de noapte, setand culoarea luminii la ("0.0f, 0.0f, 0.0f)
- Tasta "Q" – roteste scena in jurul luminii la stanga
- Tasta "E" – roteste scena in jurul luminii la dreapta
- Tasta "J" – roteste lumina in jurul scenei la stanga
- Tasta "I" – roteste lumina in jurul scenei la dreapta
- Tasta "1" – activeaza modul de vizuare pe wireframe
- Tasta "2" – vizualizarea scenei in modul solid
- Tasta "Z" – vizualizarea scenei in modul smooth
- Tasta "3" – vizualizarea scenei prin puncta
- Tasta "W" – miscarea camerei inainte
- Tasta "A" – miscarea camerei la stanga
- Tasta "S" – miscarea camerei in spate
- Tasta "D" – miscarea camerei la dreapta
- Tasta "O" – deschiderea usii
- Tasta "C" – inchiderea usii
- Tasta "0" – aprinderea farurilor masinii
- Tasta "9" – stingerea farurilor masinii

Cateva cadre din scena sunt prezentate in figurile de mai jos



Figure 3 Cadru piscina



Figure 4 Cadru interior locuinta



Figure 5 Cadru exterior locuinta

6. Concluzii si dezvoltari ulterioare

Ca si dezvoltari ulterioare se mai poate lucra la complexitatea scenei adaugand si alte obiecte 3D. De asemenea, unele texture ar putea sa fie mai bine asezate pe modele, astfel incat sa nu se extinda textura. Un alt aspect care putea fi imbunatatit ar fi modificarea skybox-ului

atunci cand trec pe modul zi. Pentru ca scena sa para cat mai realista s-ar putea adauga si alte animatii diferite de ce am implementat eu.

In concluzie, acest proiect a fost un proiect care mi-a placut foarte mult pentru ca, pe langa faptul ca am invatat cum sa folosesc libraria OpenGL, am invatat si sa modelez obiecte 3D. Mi-ar fi placut sa am mai mult timp pentru a face o scena mai complexa.

7. Referinte

[1] Laborator Prelucrare Grafica.

[2] LearnOpenGL. <https://learnopengl.com/>

Tabelul figurilor

Figure 1 Ierarhia de clase	5
Figure 2 Shadere	5
Figure 3 Cadru piscina.....	6
Figure 4 Cadru interior locuinta	7
Figure 5 Cadru exterior locuinta	7