



PRELUCRARE GRAFICĂ

Cabană în munți

Grupa 302310

Sorina - Rodica Iojă

Contents

| | |
|---|---|
| 1. Prezentarea temei..... | 2 |
| 2. Scenariul..... | 2 |
| 2.1. Descrierea scenei și a obiectelor..... | 2 |
| 2.2. Funcționalități | 3 |
| 3. Detalii de implementare..... | 3 |
| 3.1. Funcții și algoritmi | 3 |
| 3.1.1. Soluții posibile | 3 |
| 3.1.2. Motivarea abordării alese..... | 4 |
| 3.2. Modelul grafic..... | 4 |
| 3.3. Structuri de date | 4 |
| 3.4. Ierarhia de clase..... | 4 |
| 4. Prezentarea interfeței grafice | 5 |
| 5. Concluzii și dezvoltări ulterioare | 6 |
| 6. Referințe | 7 |

1. Prezentarea temei

Proiectele au ca și scop realizarea unei prezentări fotorealiste a unor scene de obiecte 3D utilizând librăriile prezentate la laborator (OpenGL, GLFW, GLM, etc.). Utilizatorul trebuie să aibă posibilitatea de a controla scena prin intermediul mausului și tastaturii.

- **(2p)** vizualizarea scenei: scalare, translație, rotație, mișcarea camerei
 - utilizând tastatura sau mausul
 - utilizând animații de prezentare
- **(1p)** specificarea surselor de lumină (cel puțin două surse de lumină diferite)
- **(0.5p)** vizualizare scenă în modurile solid, wireframe, poligonal
- **(1p)** maparea texturilor și definirea materialelor
 - calitatea texturilor și nivelul de detaliu al acestora
 - maparea texturilor pe obiecte
- **(1p)** exemplificarea generării umbrelor
- **(0.5p)** exemplificarea animării diferitelor componente ale obiectelor
- **(3p)** fotorealism, complexitatea scenei, nivelul de detaliere al modelării, dezvoltarea diferiților algoritmi și implementarea acestora (generare dinamică de obiecte, detecția coliziunilor, generarea umbrelor, ceață, ploaie, vânt), calitatea animațiilor, utilizarea diferitelor surse de lumină (globală, locală, de tip spot)
- **(1p)** documentația (obligatorie)

2. Scenariul

2.1. Descrierea scenei și a obiectelor

Scena realizată surprinde o cabană într-o zona de munte mai joasă care a fost surprinsă de venirea iernii. Cabana este înconjurată de brazi, un om de zapadă dar și lemne și loc pentru foc. Interiorul cabanei este încăpător și util fiind o canapea, un scaun suspendat, masă, perne, semineu, o lampă, dar și o scară de acces spre pod, unde se află un pat matrimonial și o lampă.

Obiecte utilizate:

- Cabana (un cub)
- Brazi
- Om de zapada
- Foc si lemne
- Skydom
- Pamant si apa
- Masa
- Carti
- Bol

- Lampa
- Canapea
- Perne
- Scaun suspendat
- Usa
- Perete de geam
- Semineu
- Scara
- Tablouri
- Pat si asternuturi
- Covor
- Parchet
- Lampa in pod
- Carbuni
- Nori

2.2. Funcționalități

- Posibilitatea navigarii cu ajutorul mouse-ului si tastelor din perspectiva first-person
- Interactiunea cu scena prin apasarea butoane poate alege unul din modurile de prezentare: solid, wireframe ,point
- Interactiunea cu un obiect care se poate translata si roti prin comenzi de la tastatura
- Sursa de lumina bidirectionala gestionabila de la tastatura
- Scena are ceata, vizibila atunci cand obiectele sunt la o distanta relativ mai mare de punctul de observatie
- O animatie de prezentare a mediului cabanei.
- Nori de furtuna
- Umbre

3. Detalii de implementare

3.1. Funcții și algoritmi

3.1.1. Soluții posibile

Importarea obiectelor cu extensia .obj din fisier, se face folosind glmModel. Acest model graphic se obtine prin identificarea in fisier a coordonatelor varfurilor triunghiurilor care il compun si reconstituirea lui pe baza acestora.

Texturile folosite in scena sunt citite din fisiere .png si pastrate intr-un fisier. Sunt apoi incarcate in proiect prin .mtl.

Functia `mouseCallback()` ajuta la implementarea miscarii camerei cu ajutorul mouseului. Functia `processMovement()` ajuta la implementarea actiunilor posibile cu ajutorul tastelor.

3.1.2. Motivarea abordării alese

Implementarea a avut la baza materialele din cadrul laboratorului. Abordarea este una simpla si eficienta folosind librariile si clasele prezentate mai sus, putem avea o manipulare intr-o maniera simpla si inteligibila toate attributele scenei si obiectele care o compun.

3.2. Modelul grafic

Libraria OpenGL Mathematics(GLM) care implementeaza partea matematica din spatele graficii pe calculator bazat pe OpenGL Shading Language(GLSL).

3.3. Structuri de date

Pentru realizarea calculelor am folosit structurile de date disponibile in biblioteca GLM, cum ar fi `vec<n>`, sau `mat<n>`, precum si structurile specifice OpenGL-ului cum ar fi `Gluint`, `GLFWwindow`, etc.

3.4. Ierarhia de clase

Pentru a implementa functionalitatile implementate anterior, a fost necesara includerea de headere si .cpp-uri precum:

- `Camera.hpp` – care realizeaza controlul camerei
- `Mesh.hpp` – defineste varfurile unui obiect
- `Model3D.hpp` – creeaza un nou model 3D
- `Shader.hpp` – incarca un shader
- `glm.hpp` – biblioteca pentru calculul matematic
- `GLEW.h` si `GLFW.h` – pentru functionalitate/randari
- `Camera.cpp`
- `Main.cpp`

4. Prezentarea interfeței grafice



Manual de utilizare:

Q -> rotatia scenei

E -> rotatia scenei

J -> rotatia luminii

L -> rotatia luminii

W -> misca camera in fata

S -> misca camera in spate

A-> misca camera la stanga

D- >misca camera la dreapta

U-> deschide usa

X->inchide usa

C- >incepe rotirea scaunului

Z- >opreste rotirea scaunului

O- >porneste lampa

P- >opreste lampa

V->mod wireframe

B- >mod punctiform

N->mod normal

M->harta de adancime (toggle)

SPACE -> apare o grupare de nori

ENTER -> dispar nori

5. Concluzii și dezvoltări ulterioare

Proiectul este deschis dezvoltarilor ulterioare. Ar putea aparea mai multe obiecte, umbrele mai detaliate, dar si o animatie de prezentare mai lunga care intra si in casa.

6. Referințe

[Learn OpenGL, extensive tutorial resource for learning Modern OpenGL](#)

[OpenGL Mathematics \(g-truc.net\)](#)