

MINISTERUL EDUCAȚIEI NAȚIONALE



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**DEPARTAMENTUL CALCULATOARE**

# **PROIECT LA PRELUCRARE GRAFICA**

**Realizat de: Bartalus Izabella**

# **CUPRINS**

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

## 1. Prezentarea temei

Obiectivul acestui proiect a fost realizarea unei scene fotorealiste alcatuita din obiecte 3D si care sa foloseasca biblioteca OpenGL.

OpenGL(Open Graphics Library) este o specificatie a unui standard care foloseste un API multiplatforma foarte utilizat pentru programarea componentelor grafice 2D si 3D ale programelor de calculator.

Scena pe care am implementat-o reprezinta un mic sat care are cateva case, o zona de camping (alcatuita din cateva rulote), un lac si diferite animale.

## 2. Scenariul

### 2.1. Descrierea scenei si a obiectelor

Precum am spus mai sus, scena implementata este un mic sat care se afla in jurul unor dealuri. Satul este compus din cinci case, un turn de veghe si mai multe rulote, unele aflate in zona de camping si inca una langa lac. Unele case sunt ingradite cu garduri.

Totodata avem mai multe tipuri de copaci, lemne, o banca, mai multe roabe, wc-uri, doua masini, mai multe tractoare, stalpi de iluminat, fantane si locuri de foc.

Se pot observa mai multe tipuri de animale, dintre care amintim: trei tipuri de caini, doua tipuri de caprioare, ursi si rate.

Toate cele enumerate mai sus se afla sub unui cer.





## 2.2. Functionalitati

Functionalitatile pe care am reusit sa le implementez sunt urmatoarele:

- Posibilitatea de a naviga prin scena(scalare, translatie, rotatie, miscarea camerei) utilizand tastatura sau mouse-ul; <sup>1</sup>
- Animatia de prezentare a scenei utilizand tastatura;
- Vizualizarea scenei in modurile solid, wireframe si poligonal apasand anumite taste;

Imagine care ilustreaza modul solid:



<sup>1</sup> Laborator 5 Prelucrare grafica



Imagine care ilustreaza modul wireframe:



Imagine care ilustreaza modul poligonal:



- Existenta mai multor tipuri de lumina care apar la apasarea anumitor taste: lumina globala, de tip punctiforma si de tip spot;<sup>23</sup>

Imagine care ilustreaza lumina punctiforma pe casa din stanga, pe una dintre rulotele din zona de camping si pe ursul din fata rulotei de la lac:



Lumina punctiforma pe ursul de langa rulota de la lac si pe una dintre rulote din zona de camping:



<sup>2</sup> Laborator 7 Prelucrare Grafica

<sup>3</sup> Laborator 8 Prelucrare Grafica



Imaginea care ilustreaza lumina de tip spot pozitionata pe felinarul din curtea casei din dreapta-sus:



- Existenta efectului de ceata, care se porneste/opreste de la tastatura, si, de asemenea, se poate regla densitatea cetii (se poate creste/scadea) tot utilizand anumite taste; <sup>4</sup>

Imagine care ilustreaza efectul de ceata:



---

<sup>4</sup> Laborator 12 Prelucrare Grafica

### 3. Detalii de implementare

#### 3.1. Funcții și algoritmi

##### 3.3.1 Soluții posibile

Libraria OpenGL cuprinde o mare varietate de functii, dintre care amintim:

- `glBindTexture()` ;
- `glGenTextures()` ;
- `glViewport()` ;
- `glfwCreateWindow()` ;
- `glTexImage2D()` etc.

Funcțiile pe care le-am folosit în proiect sunt următoarele:

- **`initShaders()`** – unde importam shaderele în program;
- **`initObjects()`** – unde importam scena;
- **`initUniforms()`** – unde setăm luminile (de tip spot, de tip punctiformă, direcțională);
- **`renderScene()`** – unde trimitem datele către shader;
- **`processMovement()`** – unde punem funcționalități pe taste pentru diferite efecte din program;
- **`mouseCallback()`** – unde implementăm funcționalitatea de mișcare a camerei folosind mouse-ul;
- **`windowResizeCallback()`** – unde implementăm funcționalitatea de a redimensiona ecranul;
- **`keyboardCallback()`** - unde implementăm funcționalitatea de mișcare a camerei folosind tastatura;

Pentru a crește realismul scenei, am adăugat funcționalitatea de ceață care va crea o altă modalitate de a putea vedea scena. Pentru a reuși să implementez ceața, am folosit documentația prezentată în laboratorul 12 de Prelucrare Grafică, de unde am ales să implementez ceața de tip exponențială patratică, care are următoarea formulă:

$$fogFactor = e^{-(fragmentDistance * fogDensity)^2}$$

Ca și tipuri de lumină am folosit în primul rând lumina globală care are trei componente: lumina ambientală (nu vine dintr-o anumită direcție, fiind o aproximare decentă a luminii care există împrăștiată în jurul unei scene. Calculul său nu depinde de poziția spectatorului sau de direcția luminii. Valoarea iluminării ambientale poate fi precalculată ca efect global sau adăugată independent pentru fiecare sursă de lumină), lumina difuză (împrăștiată în mod egal în toate direcțiile pentru o sursă de lumină. Cantitatea de lumină difuză care există pe un punct nu



depinde de poziția spectatorului, dar depinde de direcția luminii. Intensitatea luminoasă este mai puternică pe suprafețele care sunt orientate spre sursa de lumină și mai slabă pe cele orientate în direcția opusă. Calculul luminii difuze depinde de normala suprafeței și de direcția sursei de lumină, dar nu și de direcția de vizualizare) și lumina speculară (lumina reflectată direct de către suprafață și se referă la cât de asemănătoare este suprafața (materialul) obiectului cu o oglindă. Intensitatea acestui efect este denumită Shininess (strălucire). Calculul său necesită să se știe cât de apropiată este orientarea suprafeței de reflecția dintre direcția luminii și ochi. Astfel, calculul luminii speculare depinde de normala suprafeței, de direcția luminii și de direcția de vizionare).

Al doilea tip de lumină prezent în proiect este lumina de tip spot, care este o sursă de lumină care se află undeva în mediul înconjurător care, în loc să tragă raze de lumină în toate direcțiile, le trage doar într-o anumită direcție. Rezultatul este că doar obiectele aflate într-o anumită rază a direcției reflectoarelor sunt aprinse și totul rămâne întunecat. Pentru a arata un bun exemplu de lumină de tip spot, aceasta a fost pusă pe un stalp de iluminare din scenă.<sup>5</sup>

Al treilea tip de lumină prezentată în cadrul proiectului este lumina punctiformă care este o sursă de lumină cu o anumită poziție undeva în scenă în toate direcțiile, unde razele de lumină se estompează la distanță.

### **3.3.2. Motivarea abordării alese**

Am ales să realizez acest proiect deoarece tot ce am implementat porneste de la laboratoarele făcute la materia Prelucrare Grafică. Deoarece la laborator am realizat toate funcționalitățile pe un singur obiect sau pe părți de scenă nu atât de complexe, mi s-a părut interesant să încerc să implementez toate aceste funcționalități pe o scenă mai mare.

## **3.2. Modelul grafic**

Obiectele și texturile care sunt prezente în proiect au fost descărcate de pe diferite site-uri de pe internet. După ce am reușit să aplic câte o textură pentru fiecare obiect, a fost nevoie să îl scalez, translatez, roteșc în diferite moduri pentru a-l plasa corect în scenă.

## **3.3. Structuri de date**

Structurile de date pe care le-am implementat au fost cele folosite pentru a împreuna tipurile de lumină, precum cea de tip spot și cea punctiformă.

---

<sup>5</sup> [LearnOpenGL - Light casters](#)

### 3.4. Ierarhia de clase

Clasele folosite in proiect sunt urmatoarele:

- Camera.cpp – unde sunt implementate miscarile camerei;
- main.cpp – unde am implementat toate functionalitatile mentionate mai sus;
- Mesh.cpp – care reprezinta un obiect 3D;
- Model3D.cpp – contine metode pentru afisarea de mesh-uri folosind un program pentru shadere specificat;
- Shader.cpp – contine metode pentru crearea si activarea programelor shader.

## 4. Manual de utilizare

Utilizatorul poate interactiona cu scena folosind urmatoarele taste de la tastatura:

- Z – modul solid;
- X – modul wireframe;
- C – modul poligonal;
- W – miscarea camerei inainte;
- S – miscarea camerei inapoi;
- A – miscarea camerei spre stanga;
- D – miscarea camerei spre dreapta;
- R – rotatia camerei inspre dreapta;
- T – rotatia camerei inspre stanga;
- P – incepe previzualizarea camerei;
- L – opreste previzualizarea camerei;
- U – porneste efectul de ceata;
- I – intensifica densitatea cetii;
- O – scade densitatea cetii;
- Y – opreste efectul de ceata;
- V – aprinde o lumina de tip spot;
- B – stinge lumina de tip spot;
- N – aprinde o lumina punctiforma;
- M – stinge lumina punctiforma;

## 5. Concluzii si dezvoltari ulterioare

In concluzie, in cadrul proiectului realizat la materia Prelucrare Grafica am reusit sa implementez anumite functionalitati discutate in acest semestru la laborator, dintre care amintim miscarea camerei, tipurile de lumina, efectul de ceata etc.

Ca si dezvoltari ulterioare as aminti:

- Exemplificarea generarii umbrelor;
- Exemplificarea animarii diferitelor componente ale obiectelor;
- Implementarea mai multor tipuri de efecte cum ar fi: ploaie, vant, ninsoare etc.

## 6. Referințe

- Laborator 1-12 Prelucrare Grafica
- [LearnOpenGL - Light casters](#)