



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**DOCUMENTAȚIE PROIECT:**  
**SERENITATE**

STUDENT: SABĂU OANA-MARIA

DISCIPLINĂ: PRELUCRARE GRAFICĂ

GRUPA: 30232

AN UNIVERSITAR: 2023/2024



# UNIVERSITATEA TEHNICĂ

## DIN CLUJ-NAPOCA

## Cuprins

<b>1. Prezentarea temei .....</b>	<b>3</b>
<b>2. Scenariu .....</b>	<b>3</b>
<b>2.1 Descrierea scenei și a obiectelor .....</b>	<b>3</b>
<b>2.2 Funcționalități .....</b>	<b>4</b>
<b>3. Detalii de implementare .....</b>	<b>8</b>
<b>4.1 Funcții și algoritmi .....</b>	<b>8</b>
4.1.1 Soluții posibile.....	8
4.1.2 Motivarea abordării alese .....	9
<b>4.2 Modelul grafic.....</b>	<b>9</b>
<b>4.3 Structuri de date .....</b>	<b>9</b>
<b>4.4 Ierarhia de clase.....</b>	<b>10</b>
<b>4. Manual de utilizare .....</b>	<b>11</b>
4.1 Utilizarea sistemului periferic (mouse, tastatură).....	11
<b>5. Concluzii și dezvoltări ulterioare .....</b>	<b>11</b>
<b>6. Referințe .....</b>	<b>12</b>
<b>7. Index de imagini .....</b>	<b>13</b>



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

## 1. Prezentarea temei

OpenGL (Open Graphics Library) este o interfață de programare a aplicațiilor (API) pentru redarea de grafică vectorială 2D și 3D în mai multe limbaje și pe mai multe platforme. API-ul este utilizat de obicei pentru a interacționa cu o unitate de procesare grafică (GPU), pentru a obține o redare accelerată de hardware. [\[1\]](#)

Blender este o suită de creație 3D gratuită și open source. Aceasta suportă toate etapele procesului 3D - modelare, rigging, animație, simulare, randare, compoziție și urmărire a mișcării, chiar și editare video și creare de jocuri. [\[2\]](#)

Obiectivul acestui proiect a fost de a realiza o scenă realistă alcătuită din obiecte 3D prin intermediul programului Blender și încărcarea acestora în IDE-ul Visual Studio 2022 pentru a putea efectua acțiuni asupra acestora (mișcare, animații, aplicare diferiți algoritmi și proprietăți) cu ajutorul bibliotecii OpenGL.

Proiectul meu, denumit *Serenitate*, prezintă o scenă care cuprinde mai multe elemente idilice și feerice unei serii de vară (de aici și denumirea aleasă) ce înfățișează un loc liniștit înconjurat de natură – acasă. Printre elemente amintim casa părintească, animalele de companie, copacii din spatele casei, lacul cu lebede și nuferi, florile de la geam.

## 2. Scenariu

### 2.1 Descrierea scenei și a obiectelor

Scena înfățișează un peisaj romantic, de seară, desprins din poeziile eminesciene: *acasă*. Toată liniștea din lume s-a adunat în această scenă. Scena cuprinde un peisaj din natură unde se află o casă înconjurată de munte, dealuri și de copaci. O putem numi casă părintească deoarece în jurul ei se găsesc animalele de companie (câinele și pisica), florile ce aduc culoare, lacul cu nuferi pe care se plimbă liniștit lebedele. Mai întâlnim o pasăre călătoare ce vrea să își ia zborul, dar și persoana proaspăt revenită în locurile natale, femeia ce stă pe bancă contemplând peisajul ce i-a lipsit atât de mult și bucurându-se de liniște și libertate. În toată simplitatea ei, scena realizată stârnește emoții puternice privitorului. Obiectele au fost descărcate de pe site-urile *Free3D* [\[3\]](#) și *Turbosquid* [\[4\]](#), unele au venit deja gata texturate, iar pentru altele am creat texturi proprii. Planul a fost realizat urmărind indicațiile din tutorialele oferite de către îndrumătorul de laborator. [\[5\]](#)



Figura 1. Prezentarea scenei

## 2.2 Funcționalități

Funcționalitățile implementate sunt următoarele:

- posibilitatea de a vizualiza scena (efectuarea operațiilor de translație, rotație, mișcarea camerei) utilizând tastatura și mouse-ul;
- prezentarea inițială a scenei prin intermediul unei animații de prezentare;
- specificarea unei surse de lumină (pozițională);
- vizualizarea scenei în modurile solid, wireframe, poligonal și smooth;



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA



Figura 2. Vizualizarea scenei în modul solid/smooth



Figura 3. Vizualizarea scenei în modul wireframe



Figura 4. Vizualizarea scenei în modul poligonal

- obiectele din scenă sunt mapate cu texturi calitative și conferă fotorealism: în figura de mai jos se poate observa cât de complexe sunt obiectele, fiind compuse dintr-un număr considerabil de puncte, arătând ca și cum ar fi în modul solid sau wireframe;

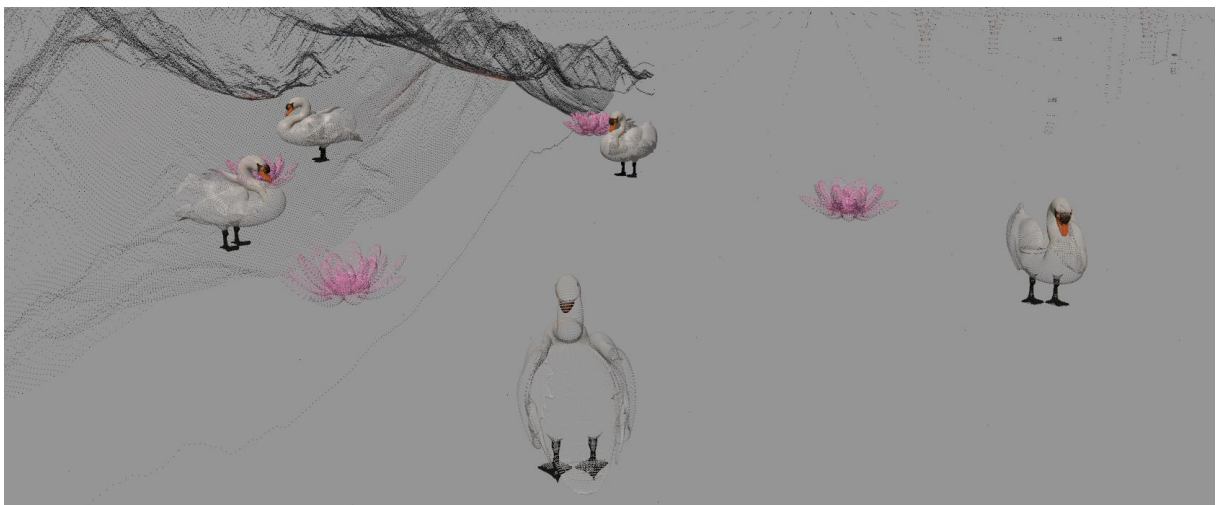


Figura 5. Evidențierea complexității obiectelor din scenă



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

- generarea umbrelor;



Figura 6. Evidențierea umbrelor

- activarea ceții folosind tastatura.



Figura 7. Prezența ceții în scenă





# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

## 3. Detalii de implementare

În acest capitol voi prezenta modul în care am ales să implementez proiectul și fundamentele de la care am pornit.

### 4.1 Funcții și algoritmi

#### 4.1.1 Soluții posibile

Pornind de la cerințele impuse, am încercat să implementez cât mai multe dintre acestea. Pot spune că proiectul meu conține exclusiv funcționalitățile discutate în cadrul laboratorului, la celălalte întâmpinând dificultăți. Voi prezenta pe scurt soluțiile posibile alese.

Am implementat o sursă de lumină pozițională, anume cea discutată în cadrul laboratorului 8 [\[6\]](#). Aceasta luminează elementele din scena mea și se rotește folosind anumite taste și servește la generarea umbrelor.

Umbrele au fost adăugate folosind tehnica *Shadow Mapping* (hărți de umbre), discutată în cadrul laboratorului 9 [\[7\]](#). Umbrele conferă fotorealism, fiind un fenomen inevitabil întâlnit în viața reală când o sursă de lumină lovește un element. Shadow Mapping este o tehnică multi-trecere care utilizează texturi de adâncime pentru a decide dacă un punct se află în umbră sau nu, precum se descrie în îndrumătorul de laborator. Algoritmul este compus din 2 etape, anume rasterizarea scenei din punctul de vedere al luminii (se creează harta de adâncime) și rasterizarea scenei din punctul de vedere al observatorului (poziția camerei). Avantajul folosirii OpenGL este că acesta face toată rasterizarea (informații despre culoare, adâncime și șablon) într-un framebuffer. Realizarea cerințelor de la laboratorul 9 mi-a facilitat munca la proiect deoarece am putut utiliza aceleași programe shader fără a le modifica și pentru obiectele mele.

Laboratorul 10, cel care prezintă ideea de Skybox nota, a servit la atribuirea unui skybox potrivit proiectului meu pentru a se încadra cu tematica aleasă. [\[8\]](#)

În ultimul laborator [\[9\]](#) am implementat și fenomenul de ceață, iar prin intermediul unei variabile booleene numită *fog* controlez dacă scena finală va avea sau nu aplicată ceață (în shaderul *shaderStart.frag*), astfel decizia de a avea în scenă ceață sau nu îi aparține utilizatorului.



#### 4.1.2 Motivarea abordării alese

Implementarea proiectului a pornit de la rezolvarea din cadrul laboratoarelor de Prelucrare Grafică, ducând la bun sfârșit cerințele propuse în fiecare săptămână deoarece știam că îmi vor servi la realizarea lui. Am ales această abordare deoarece am folosit cunoștințele acumulate de-a lungul semestrului și pot spune că am lucrat la proiect gradual.

#### 4.2 Modelul grafic

Acesta s-a obținut folosind editorul de obiecte 3D numit Blender, acesta permițând numeroase operații pentru construirea și animarea scenelor, fiind o unealtă folosită în dezvoltarea jocurilor video. Obiectele au fost descărcate de pe internet, unele au venit însoțite de texturi, iar pentru altele am creat texturi proprii. Prin intermediul Blender am construit scena finală, la indicațiile tutorialelor puse la dispoziție de către îndrumătorul de laborator, Cosmin Nandra. Obiectele au fost scalate, translatate și rotite tot în Blender și a reprezentat un avantaj mare: dacă făceam aceste operații direct în Visual Studio, era o muncă îndelungată și ineficientă.

#### 4.3 Structuri de date

Am folosit exclusiv structurile de date incluse în template-ul proiectului deoarece nu am avut nevoie de structuri de date adiționale, însă pentru implementarea luminilor punctiforme (ceea ce nu am dus la bun sfârșit) se puteau crea structuri de date specifice care să stocheze informații relevante despre acestea (factorii de ambianță, difuzie, poziție etc.).

#### 4.4 Ierarhia de clase

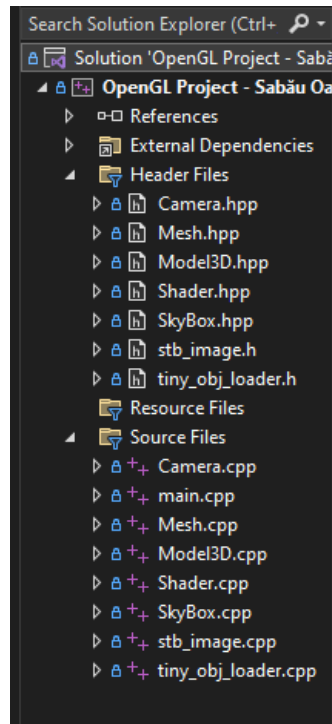


Figura 8. Ierarhia de clase a programului

Între clasele date, putem aminti funcționalitatea unora dintre acestea:

- Camera.cpp – sunt implementate mișcările camerei;
- Mesh.cpp – reprezintă un obiect 3D;
- Shader.cpp – conține metode pentru crearea și activarea programelor shader;
- main.cpp – conține implementarea proiectului și a funcționalităților amintite mai sus.



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

## 4. Manual de utilizare

### 4.1 Utilizarea sistemului periferic (mouse, tastatură)

Prin intermediul mouse-ului și a tastaturii putem efectua vizualizarea scenei create, făcând operațiile de translație, scalare, rotație, mișcarea și poziționarea camerei. Prin intermediul tastaturii ne putem deplasa doar în planul 2D, mouse-ul permițând rotirea camerei la 360°.

Utilizatorul poate interacționa cu scena folosind următoarele taste:

- W – deplasarea înainte
- A – deplasarea spre stânga
- S – deplasarea înapoi
- D – deplasarea spre dreapta
- J – rotirea sursei de lumină în sens negativ
- L – rotirea sursei de lumină în sens pozitiv
- F – activarea/dezactivarea ceții
- V – comutarea între modurile de vizualizare (solid, wireframe, poligonal) prin apăsări succesive

## 5. Concluzii și dezvoltări ulterioare

Acest proiect prezintă o scenă 3D în care am implemenat anumite funcționalități discutate de-a lungul semestrului în cadrul laboratorului de Prelucrare Grafică și a constituit fundamentul învățării și înțelegerii graficii pe calculator, un domeniu vast, dar foarte interesant și important.

Proiectul poate fi extins, în primul rând, prin extinderea scenei (adăugarea mai multor elemente – dar deja ilustrăm o altă idee și ne îndepartăm de tema aleasă), adăugarea mai multor tipuri de lumină sau a mai multor animații (implicit a animării diferitelor componente ale unui obiect). O altă extindere poate fi reprezentată de implementarea diferiților algoritmi (detectare a coliziunilor etc) sau a diferitelor efecte metereologice, adăugarea unor funcții care schimbă skybox-ul în funcție de momentul zilei etc.



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Per total, a fost un proiect în a cărui implementare am întâmpinat dificultăți din cauza faptului că sunt novice în utilizarea editorului Blender, dar rezultatul final justifică efortul depus.

## 6. Referințe

- [1] <https://en.wikipedia.org/wiki/OpenGL>
- [2] <https://www.blender.org/about/>
- [3] <https://free3d.com/>
- [4] <https://www.turbosquid.com/>
- [5] [https://www.youtube.com/playlist?list=PLrgcDEgRZ\\_kndoWmRkAK4Y7ToJdOf-OSM&themeRefresh=1](https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM&themeRefresh=1)
- [6] [https://moodle.cs.utcluj.ro/pluginfile.php/186372/mod\\_resource/content/2/Laboratory\\_work\\_8\\_RO.pdf](https://moodle.cs.utcluj.ro/pluginfile.php/186372/mod_resource/content/2/Laboratory_work_8_RO.pdf)
- [7] [https://moodle.cs.utcluj.ro/pluginfile.php/186388/mod\\_resource/content/4/Laboratory%20work%209%20RO.pdf](https://moodle.cs.utcluj.ro/pluginfile.php/186388/mod_resource/content/4/Laboratory%20work%209%20RO.pdf)
- [8] [https://moodle.cs.utcluj.ro/pluginfile.php/186397/mod\\_resource/content/3/Laboratory\\_work\\_10%20RO.pdf](https://moodle.cs.utcluj.ro/pluginfile.php/186397/mod_resource/content/3/Laboratory_work_10%20RO.pdf)
- [9] <https://www.ambiera.com/forum.php?t=7873>



# UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

## 7. Index de imagini

Figura 1. Prezentarea scenei.....	4
Figura 2. Vizualizarea scenei în modul solid/smooth.....	5
Figura 3. Vizualizarea scenei în modul wireframe.....	5
Figura 4. Vizualizarea scenei în modul poligonal.....	6
Figura 5. Evidențierea complexității obiectelor din scenă.....	6
Figura 6. Evidențierea umbrelor.....	7
Figura 7. Prezența ceții în scenă.....	7
Figura 8. Ierarhia de clase a programului.....	10