

MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE



---

**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**  
**CATEDRA CALCULATOARE ȘI TEHNOLOGIA**  
**INFORMATIEI**

# **Prelucrare Grafică**

## **Documentație Proiect**

**Indrumator de proiect: Sabou Adrian**

**Student: Morar Timotei Cristian**

**Data: 17.01.2024**

**Grupa: 30235**

## CUPRINS

1. PREZENTAREA TEMEI .....	3
2. SCENARIUL.....	3
Descrierea scenei si a obiectelor.....	3
Functionalitati .....	4
3. DETALII DE IMPLEMENTARE .....	5
Functii si algoritmi .....	5
Modelul grafic .....	6
Structuri de date .....	6
Ierarhii de clase.....	7
4. PREZENTAREA INTERFETEI GRAFICE .....	7
5. MANUAL DE UTILIZARE .....	8
6. CONCLUZII .....	9
7. REFERINTE .....	9

# 1. PREZENTAREA TEMEI

Scopul acestui proiect constă în oferirea unei experiențe profunde de fotorealism în jocuri și nu numai, dezvăluind procesul de prelucrare a unei scene cu obiecte de către calculator. În același timp, proiectul își propune să instruiască utilizatorul în utilizarea diverselor biblioteci și librării 3D, cum ar fi OpenGL, GLFW și GLM. Interfața aplicației permite utilizatorului să exploreze scena atât prin intermediul tastaturii, cât și al mouse-ului, recreând senzația dintr-un joc pe calculator. Pentru a facilita această experiență, am optat pentru o scenă predefinită, concentrându-mă asupra aranjării obiectelor în modul dorit. Texturarea acestora a necesitat un efort semnificativ, dar rezultatele obținute justifică în totalitate dedicarea timpului și atenției acordate acestui aspect.

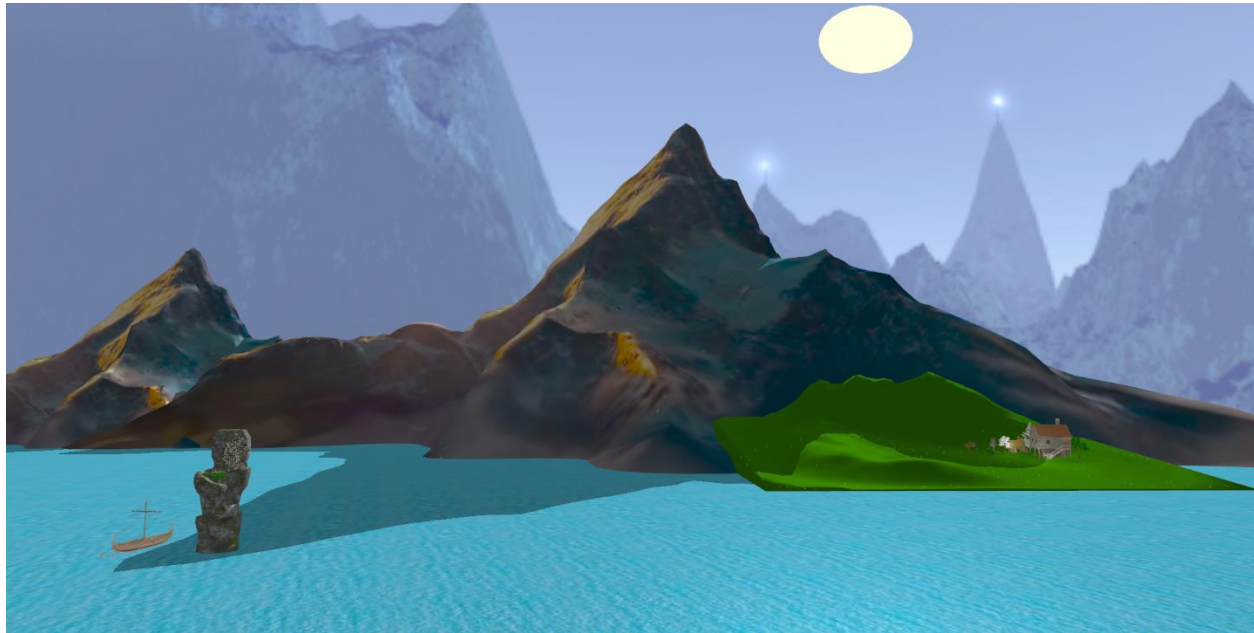
## 2. SCENARIUL

### Descrierea scenei și a obiectelor

Scena pe care am creat-o reprezintă un colț al naturii rupt dintr-un univers plin de mister, asemănător unui basm viking. Imaginea este dominată de o vastă mare cu insule pitorești și mărețe munți, sub un cer senin pe care l-am dorit să fie perfect, dar care poate fi personalizat pentru a se potrivi atmosferei dorite. Personajul principal al acestui basm găsește adăpost într-o casă pitorească, o locuință sub umbra copacilor, departe de razele soarelui. Crearea acestei scene a reprezentat o provocare, deoarece a fost necesar să texturăm cu grijă fiecare obiect și să-l plasăm strategic în locurile care mi-au captat atenția, rotindu-le pentru a oferi un aspect plăcut. Am întâmpinat dificultăți inițiale în importarea scenei în OpenGL, fiind dezamăgit de discrepanța dintre aspectul scenei în Blender și cel în Visual Studio. Implementarea fragment discarding a fost soluția salvatoare, aducând scenei un aspect autentic și captivant.

Miscarea umbrelor, împreună cu lumina direcțională, poate fi controlată de tastele Q și E, contribuind la atmosfera naturală a peisajului viking. O adiție specială

este ceața, care poate fi activată cu tasta 8, contribuind la misterul și farmecul acestei lumi pline de aventură și poveste.



## Funcționalități

Pentru a oferi o experiență vizuală captivantă, scena este iluminată de o sursă de lumină punctiformă care se mișcă prin întreaga imagine, evidențiind astfel implementarea umbrelor. Utilizatorii pot explora scena cu ajutorul tastelor W, A, S, D și a mouse-ului, oferindu-le libertatea de a se plimba în această lume fascinantă. Pentru a adăuga un element de realism, implementarea fragment discarding a fost esențială, asigurând ca detaliile precum frunzele copacilor să se integreze armonios în fundalul texturat al scenei.

Posibilitatea de a simula diferite condiții meteorologice a fost introdusă, permițând utilizatorilor să experimenteze diverse situații apăsând tastele Z, X și C. Pentru a adăuga o atmosferă misterioasă, ceața poate fi activată prin apăsarea tastei 8.

Umbrele au fost implementate folosind tehnica de shadow mapping, prezentată în laborator, deoarece aceasta este ușor de integrat în OpenGL. Rotirea luminii pentru a experimenta efectul de umbră se realizează cu tastele E și Q. Pentru a analiza harta de adâncime, este suficient să apăsăm tasta M. De asemenea, pentru a roti doar soarele și a modifica unghiul de iluminare, utilizăm tastele J și L. Toate aceste elemente contribuie la o experiență vizuală complexă și captivantă în lumea virtuală creată.

# 3. DETALII DE IMPLEMENTARE

## Funcții și algoritmi

În bibliotecile pentru OpenGL, se regăsesc numeroase funcții ajutătoare, iar prima în ordinea apariției în proiectul meu este cea dedicată inițializării ferestrei pe care se proiectează harta. Această funcție primește ca parametri dimensiunile vizuale ale ferestrei și un nume, având forma: `myWindow.Create(glWindowWidth, glWindowHeight, "OpenGL Project Core")`.

Deoarece fiecare punct de pe ecran este desenat individual și procesat în `shader`, funcția `renderScene()` este esențială în proiect. Aici, obiectele sunt desenate în funcție de parametrul `M`, unde, dacă acesta nu este apăsat, se desenează întreaga scenă, iar în caz contrar, se afișează doar harta de adâncime pentru a evalua cum sunt desenate umbrele.

O altă funcție crucială este `initUniforms()`, unde se configurează adresele locațiilor variabilelor din `shader`, inițializându-le cu valori specifice, cum ar fi detalii despre iluminare, perspectiva camerei sau matricea de normale. Fără această funcție, programul nu ar funcționa corect. De asemenea, menționez funcțiile `initModels()` și `initShaders()`, care permit încărcarea în OpenGL a modelelor 3D și a `shader`elor.

Implementarea mișcării camerei prin tastatură și mouse a necesitat funcțiile `mouseCallback()` și `processMovement()`, asigurând o vizualizare facilă a scenei.



Pentru a îmbunătăți aspectul fundalului, am introdus și un skybox, simulând infinitatea lumii, cu ajutorul funcției `renderSkyBox()`. Aceasta utilizează un skybox cu cer senin, soare pe cer și munți în depărtare, adăugând o notă de realism și extinzând perspectiva asupra lumii create.

## Modelul grafic

Harta și obiectele pe care le-am utilizat în proiect nu sunt creații originale, ci au fost descărcate de pe diverse site-uri de pe internet. Contribuția mea a constat în aranjarea și texturarea lor în Blender. Alegerea de a utiliza resurse preexistente a fost motivată de dorința de a simplifica munca mea, având în vedere că proiectul implica gestionarea a peste 100 de obiecte. Importarea fiecărui obiect individual ar fi necesitat un efort considerabil, astfel că am optat pentru o abordare mai eficientă.

Prin aranjarea și texturarea acestor obiecte în Blender, am reușit să obțin o scenă coerentă și estetică fără a sacrifica detaliile și calitatea vizuală. Această strategie a facilitat considerabil procesul creativ și a condus la rezultate realiste și atrăgătoare.

## Structuri de date

În cadrul proiectului meu, structurile de date reprezintă elemente esențiale, iar una dintre ele se regăsește în clasa dedicată camerei. Aici, lucrăm cu mai mulți vectori, precum poziția camerei, direcția de vizualizare și coordonatele camerei, elemente cruciale pentru a gestiona perspectiva și mișcarea în scenă.

O altă structură de date semnificativă este `Glint`, în care sunt stocate adresele variabilelor uniform din shading, alături de vectori de diverse dimensiuni și mărimi. Această structură joacă un rol important în gestionarea interacțiunii între programele shader și programul principal.

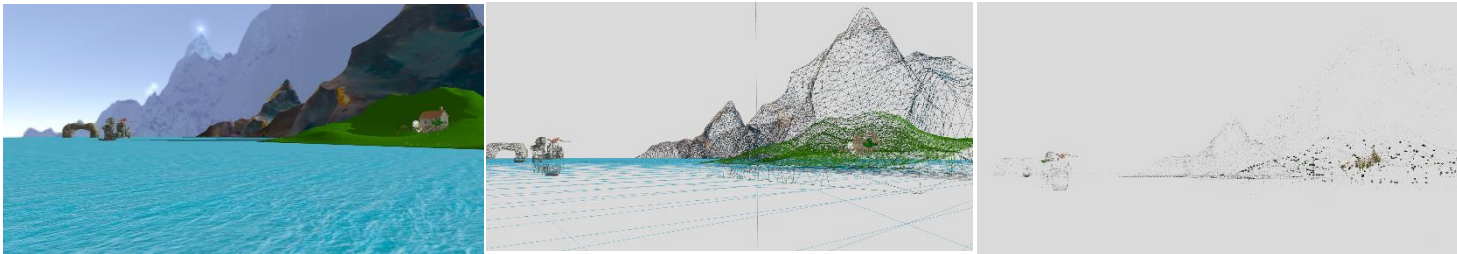
De asemenea, proiectul utilizează și structuri de date din C++ precum `Model` și `Shaders`, împreună cu tipuri de date normale precum `bool`, `int` și `char`. Aceste variabile obișnuite sunt esențiale pentru implementarea diverselor funcționalități în cadrul proiectului, aducând consistență și eficiență în gestionarea datelor.

## Ierarhii de clase

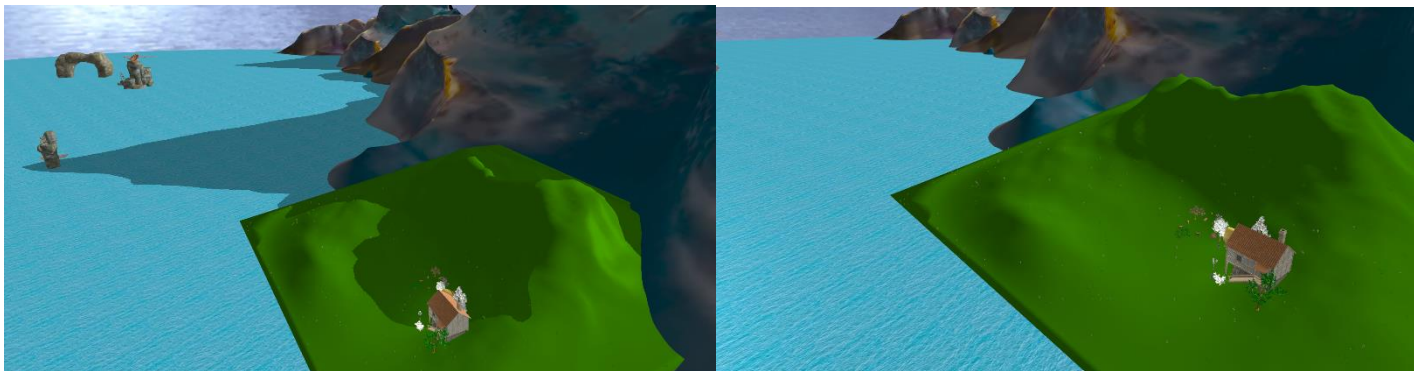
În afara de main fiecare clasă are câte un header. Urmează acum să dau exemplu de clase și fiecare ce face. Clasa de cameră are implementat în ea camera, clasa de window are implementată toate funcționalitățile pentru crearea și ștergerea unei ferestre de desenare, clasa de skyBox încarcă, desenează și efectuează diverse operații pe sky box, clasa de mesh desenează obiecte, clasa de shaders efectuează toate operațiile necesare implementării tuturor funcționalităților unui shader nu în ultimul rând std\_image și tiny\_obj\_loader texturează respectiv încarcă un obiect în OpenGL.

## 4. PREZENTAREA INTERFETEI GRAFICE

Moduri de vizualizare:

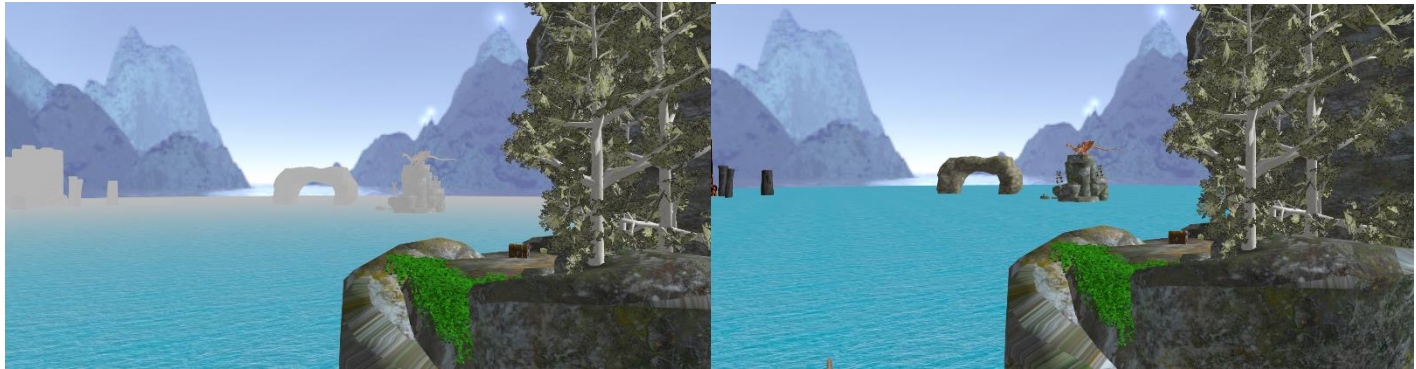


Cu și fără umbre:





Cu si fără ceata:



## 5. MANUAL DE UTILIZARE

Utilizatorul se poate bucura de urmatoarele functionalitati implementate si legate de taste:

- 8 - activarea ceții.
- W - deplasare camera în față
- S - deplasare camera în spate
- A - deplasare camera în stânga
- D - deplasare camera în dreapta
- L - deplasare lumina în cerc (În partea dreaptă)
- J - deplasare lumina în cerc (în partea stângă)
- E – rotire scena (în partea dreapta)
- Q - rotire scena (în partea stanga)
- M – afișarea hărții de adâncime
- Z – vizualizarea scenei în modul solid
- X – vizualizarea scenei în modul wireframe
- C – vizualizarea scenei în modul polygonal
- V – vizualizarea scenei în modul smooth
- R – pornește animația



## 6. CONCLUZII

Acest proiect reprezintă o resursă valoroasă pentru studenții dornici să exploreze modelarea în OpenGL, fiind adresat în special celor pasionați de jocurile pe calculator și senzațiile captivante pe care le oferă. Prin intermediul acestui proiect, am avut ocazia să învățăm tehnicile de texturare a obiectelor, modelarea unui obiect 3D și aplicarea diverselor transformări asupra acestora.

Pentru mine, a fost o experiență plăcută să dezvolt prima mea aplicație implementată în stil first-person. Îmi doresc să continuu să aprofundez aceste tehnici, conștient fiind că unele dintre ele reprezintă provocări semnificative pentru cei neinițiați în acest domeniu. Prin acest proiect, sper să îmi îmbogățesc cunoștințele și să descopăr noi aspecte ale modelării și programării în OpenGL, deschizând astfel calea către o înțelegere mai profundă a acestui fascinant domeniu.

## 7. REFERINTE

<https://learnopengl.com/> [pentru a învăța bazele pt OpenGL]

<https://free3d.com/> [pentru diferite obiecte]

<https://wow.tools/export/> [pentru a descărca diferite hărți]