

MINISTERUL EDUCAȚIEI NAȚIONALE



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

DOCUMENTAȚIE PROIECT

Procesare Grafică

SÎRGHI PAULA
GRUPA 30233

CUPRINS

1. Prezentarea temei	2
2. Scenariul	2
a) Descrierea scenei și a obiectelor	2
b) Funcționalități	4
3. Detalii de implementare	5
a) Funcții și algoritmi	5
I. Soluții posibile	5
II. Motivarea abordării alese	6
b) Modelul grafic	6
c) Structuri de date	6
d) Ierarhie de clase	6
4. Prezentarea interfeței grafice utilizator / manual de utilizare	6
5. Concluzii și dezvoltări ulterioare	9
6. Referințe	9

1. Prezentarea temei

Subiectul acestei teme este reprezentat de prezentarea fotorealismă a obiectelor 3D, folosind biblioteca OpenGL.

Scena aleasă este constituită dintr-o insulă minimalistă cu un relief muntos. Insula este străbătută de un râu care se varsă în marea ce o înconjoară. Totodată, aici pot fi observate obiecte precum: bazi, două clădiri, un pod și o barcă.

2. Scenariul

a) Descrierea scenei și a obiectelor

Scena proiectului surprinde o casă situată pe o insulă, peste care se poate observa un peisaj montan folosit pe post de SkyBox. Alte obiecte sesizabile sunt brazii din spatele casei, podul, barca, banca și o clădire-observator. Totodată, pe tot parcursul execuției programului se poate auzi lătratul câinelui.





În casă pot fi observate mai multe obiecte, cum ar fi: un pat, două noptiere, trei surse de lumină, două rafturi, scări, televizor, fotolii, o masă, o canapea, un cățel, o minge, un covor, o vază cu flori și o etajeră.



b) Funcționalități

Utilizatorul poate vizualiza întreaga scenă prin intermediul utilizării tastelor a,w,s,d,q,e și a mouse-ului. Totodată, scena poate fi privită și din perspectiva modului solid, wireframe, poligonal și smooth.

Am folosit mai multe surse de lumină, cum ar fi:

- lumina direcțională – lumina emisă de o sursă de iluminare invizibilă, situată deasupra scenei, înfinit de îndepărtată, aceasta putând fi controlată prin intermediul tastaturii. Datorită luminii direcționale, de-a lungul scenei pot fi vizualizate umbre pentru toate obiectele scenei.
- 3 lumini punctiforme de culori diferite. Acestea luminează radial și uniform în toate direcțiile, iar razele lor se estompează în funcție de distanță.



Animațiile sunt reprezentate de saltul mingii (o translație pe axa y) și de mișcarea cozii câțelului (o rotație pe axa z).

3. Detalii de implementare

a) Funcții și algoritmi

I.Soluții posibile

Biblioteca OpenGL Extension Wrangler (GLEW) este o bibliotecă de încărcare a extensiilor C/C++ open source multiplatformă. GLEW oferă mecanisme eficiente de rulare pentru a determina ce extensii OpenGL sunt acceptate pe platforma țintă.

GLFW ne va permite să creăm o fereastră și să primim input de la mouse și tastatură. OpenGL nu se ocupă de crearea sau introducerea ferestrelor, așa că trebuie să folosim aceste biblioteci pentru a gestiona ferestre, tastatură, mouse și așa mai departe.

Printre algoritmii folosiți enumerăm algoritmul Blinn-Phong, pentru crearea luminilor punctiforme.

Funcțiile folosite sunt:

- `glCheckError` – pentru a afișa erorile întâmpinate.
- `renderScene()` – pentru a trimite date shaderelor și a realiza operațiile de rotație, scalare și translație.
- `initUniforms()` – pentru a seta detaliile luminii.
- `initShaders()` – pentru inițializarea shaderelor.
- `initObjects()` – pentru încărcarea obiectelor în scenă.
- `computeLightSpaceTrMatrix()` – pentru a returna matricea de transformare a luminii în spațiu.
- `Minge()` și `tail_function()` – pentru realizarea animațiilor.
- `drawObjects()` – pentru desenarea obiectelor în scenă.
- `initSkyBox()` – pentru a încărca fețele skyBox-ului.

II. Motivarea abordării alese

Datorită influenței laboratoarelor referitoare la calcularea luminii, a umbrelor, realizarea operațiilor de translație, rotație și scalare a diferitelor obiecte, mi-a fost mult mai ușor să îmi aleg o anumită temă, folosindu-mă de scheletul unui proiect realizat în unul din laboratoare. Tema aleasă este în strânsă legătură cu pasiunea mea pentru jocurile video de tipul open world.

b) Modelul grafic

Obiectele și texturile au fost descărcate, în principal, de pe internet pentru a le putea importa în Blender și a le edita și plasa în scena finală.

c) Structuri de date

Structurile de date folosite sunt vectori, matrici și orice alte structuri necesare pentru a reține atributele luminilor specificate mai sus. Majoritatea au fost deja definite în lucrările de laborator, acest lucru ușurându-mi munca.

d) Ierarhie de clase

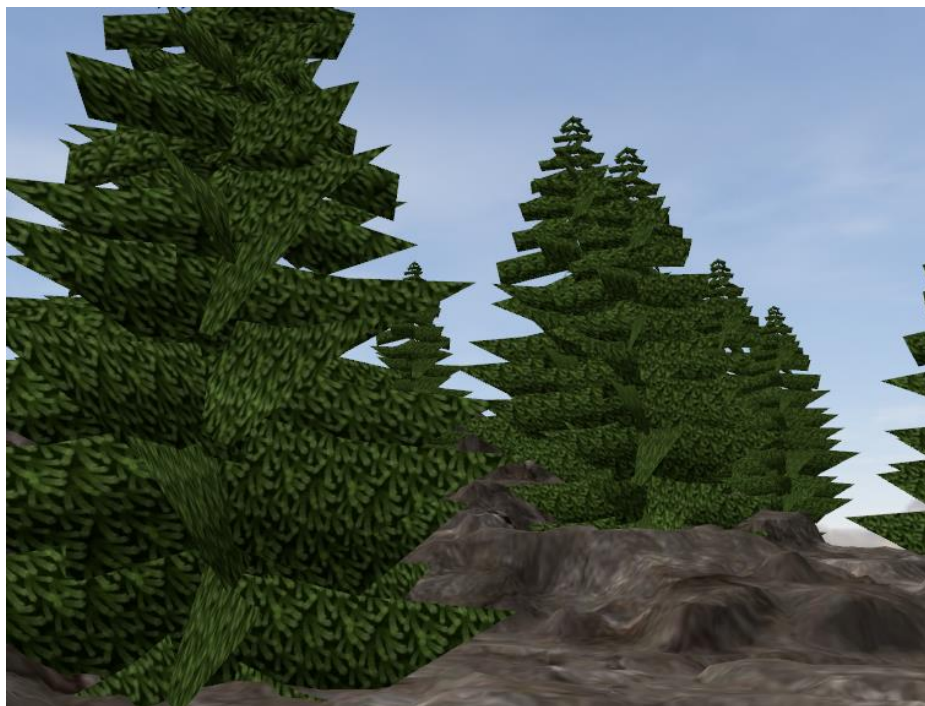
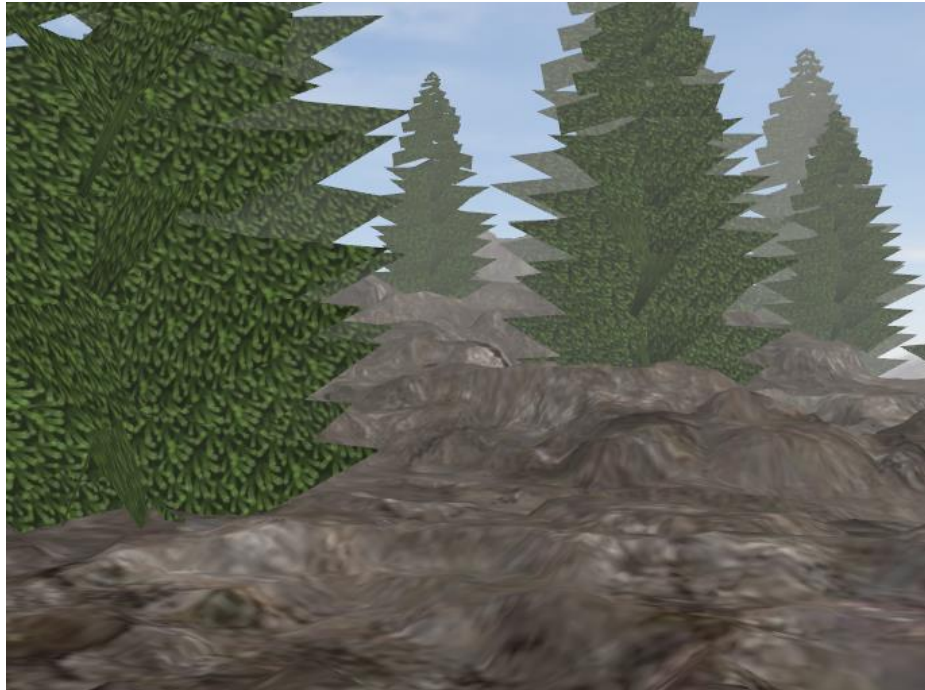
- **Camera** conține funcționalitățile camerei (mutare sus, jos, stânga, dreapta, în spate, în față și rotații).
- **SkyBox** conține implementarea skybox-ului.
- **Mesh** – pentru reprezentarea obiectelor 3D.
- **Model3D** – pentru afișarea modelelor, folosind un shader.
- **Shader** – pentru crearea și activarea shaderelor.

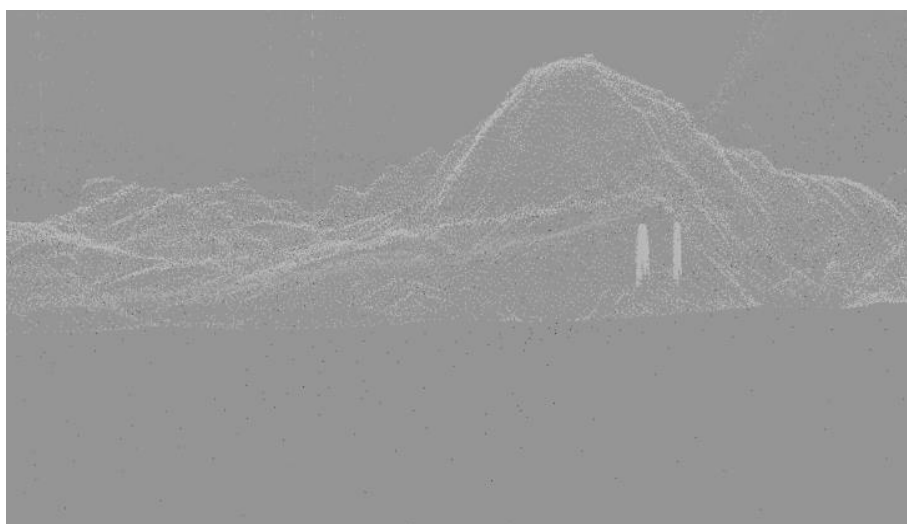
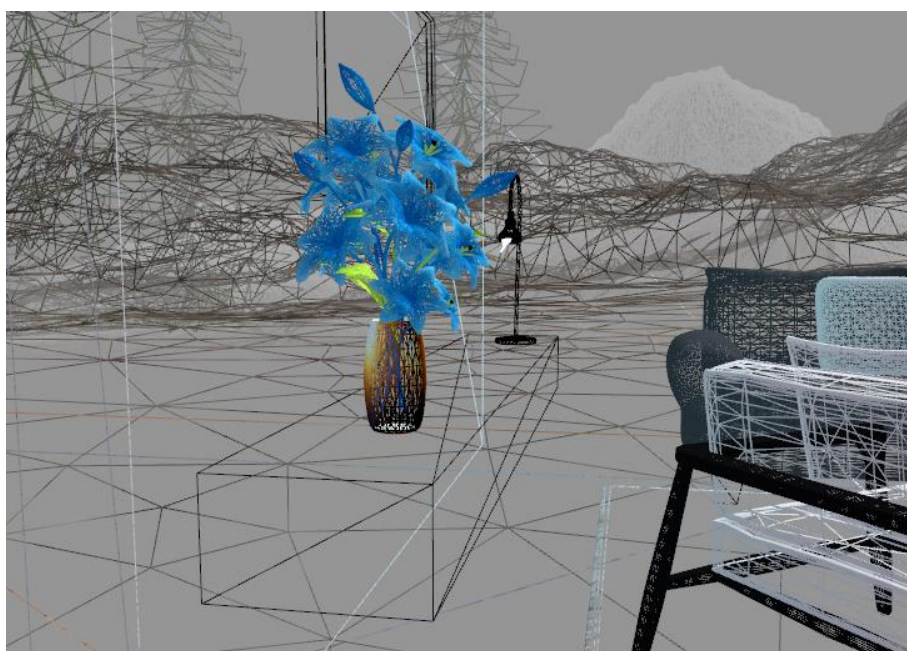
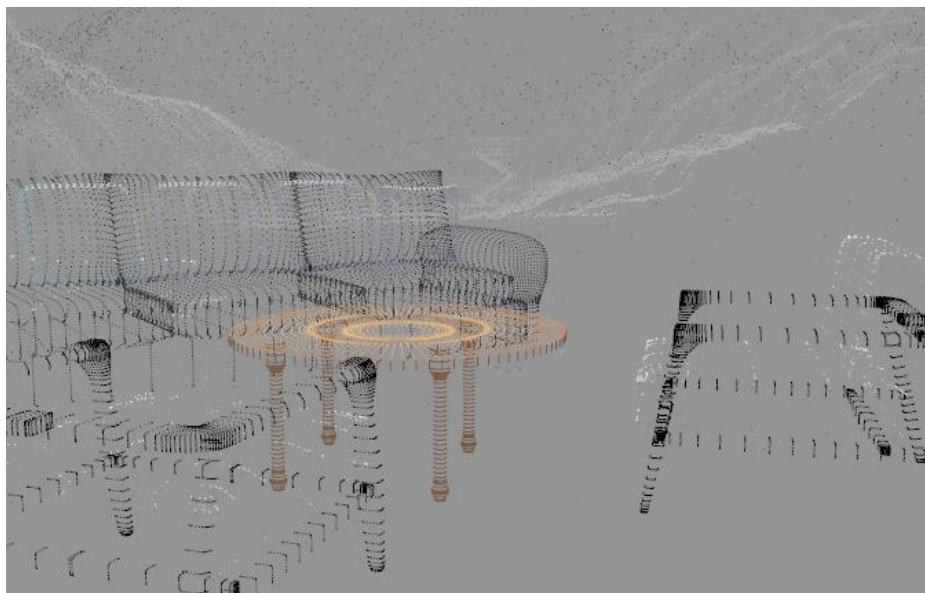
4. Prezentarea interfeței grafice utilizator / manual de utilizare

Utilizatorul dispune de numeroase taste pentru a interacționa cu obiectele din scenă:

- mouse – rotația camerei
- W / S – mutarea camerei în față / spate
- A / D – mutarea camerei în stânga / dreapta
- Q / E – rotirea camerei la stânga / dreapta
- F / G – activarea / dezactivarea ceții
- V / B – creșterea / scăderea intensității ceții
- M – animație cameră
- N – validare animație cameră

- J / K – rotația luminii cauzatoare de umbră
- Z/ X/ C – vizionarea scenei în modul solid/ wireframe/ point
- I/O – animația cozii
- 5/6 – animația mingii





5. Concluzii și dezvoltări ulterioare

În concluzie, acest proiect a fost o adevărată provocare pentru mine, nefiind obișnuită să lucrez în Blender. Deși a durat ceva până să mă obișnuiesc cu modul de lucru, pot spune că a fost o experiență destul de interesantă și totodată o provocare. Codul propriu-zis a fost destul de complicat și a necesitat o perioadă îndelungată, fiind mult mai dificil decât am speculat inițial, petrecând destul timp încercând să înțeleg mai bine unele laboratoare. Proiectul are la bază o muncă meticuloasă, ducând în final la obținerea unui cumul de experiență considerabil.

6. Referințe

- OpenGL sounds - [https://docs.microsoft.com/en-us/previous-versions/dd743680\(v=vs.85\)](https://docs.microsoft.com/en-us/previous-versions/dd743680(v=vs.85))
- Cursurile și laboratoarele de Prelucrare Grafică (D. Gorgan) -
- Tutorial Blender (C. Nandra)