

PROIECT PRELUCRARE GRAFICĂ

Irini Karina

30239

Cuprins

1. Prezentarea temei	3
2. Scenariul	3
2.1. Descrierea scenei și a obiectelor	3
2.2. Funcționalități	4
3. Detalii de implementare	6
3.1. Funcții și algoritmi	6
3.1.1. Soluții posibile	6
3.1.2. Motivarea abordării alese	8
3.2. Modelul grafic	8
3.3. Structuri de date	9
3.4. Ierarhia de clase	9
4. Prezentarea interfeței grafice utilizator	11
5. Concluzii și dezvoltări ulterioare	12
6. Referințe	13

1. Prezentarea temei

Proiectul constituie realizarea unei prezentări fotorealiste a unei scene de obiecte 3D utilizând librăriile prezentate la laborator (OpenGL, GLFW, GLM). Utilizatorul are posibilitatea de a interacționa cu aceasta prin tastatură și mouse. Proiectul abordează nu doar aspecte fundamentale precum scalarea, translația, rotația și mișcarea camerei, ci și adaugă complexitate prin implementarea animațiilor elaborate, gestionarea a cel puțin două surse de lumină, vizualizarea scenelor în moduri diverse (solid, wireframe, poligonal) și generarea umbrelor.

Alegerea unei scene arctice, mai exact o insulă, adaugă o dimensiune captivantă proiectului, oferind o paletă distinctă și posibilități de a evidenția elemente specifice acestei regiuni.

2. Scenariul

Inițial, am optat să dezvolt scena cu obiecte statice utilizând platforma Blender. Am exportat separat obiectele pentru care intenționez să aplic anumite animații. Acestea au fost ulterior încărcate în mediul de dezvoltare Visual Studio, unde am gestionat integrarea lor în scena finală a proiectului prin intermediul translațiilor și scalărilor, asigurându-ne că au dimensiunile potrivite în raport cu celelalte obiecte deja existente.

2.1. Descrierea scenei și a obiectelor

Scena dezvoltată conține:

- Teren cu munți acoperiți de zăpadă și o vale
- Skybox care reprezintă ziua și noaptea
- Lac înghețat înconjurat de pinguini și foci
- Cort de adăpostire
- Iglu cu lămpi la intrare
- Turn cu flacără

- Urși polari și reni cu sanie
- Snowmobil
- Copaci acoperiți de zăpadă, potriviți pentru atmosfera specifică scenei
- Bărbat naufragiat ajuns pe această insulă cu vaporul Titanic

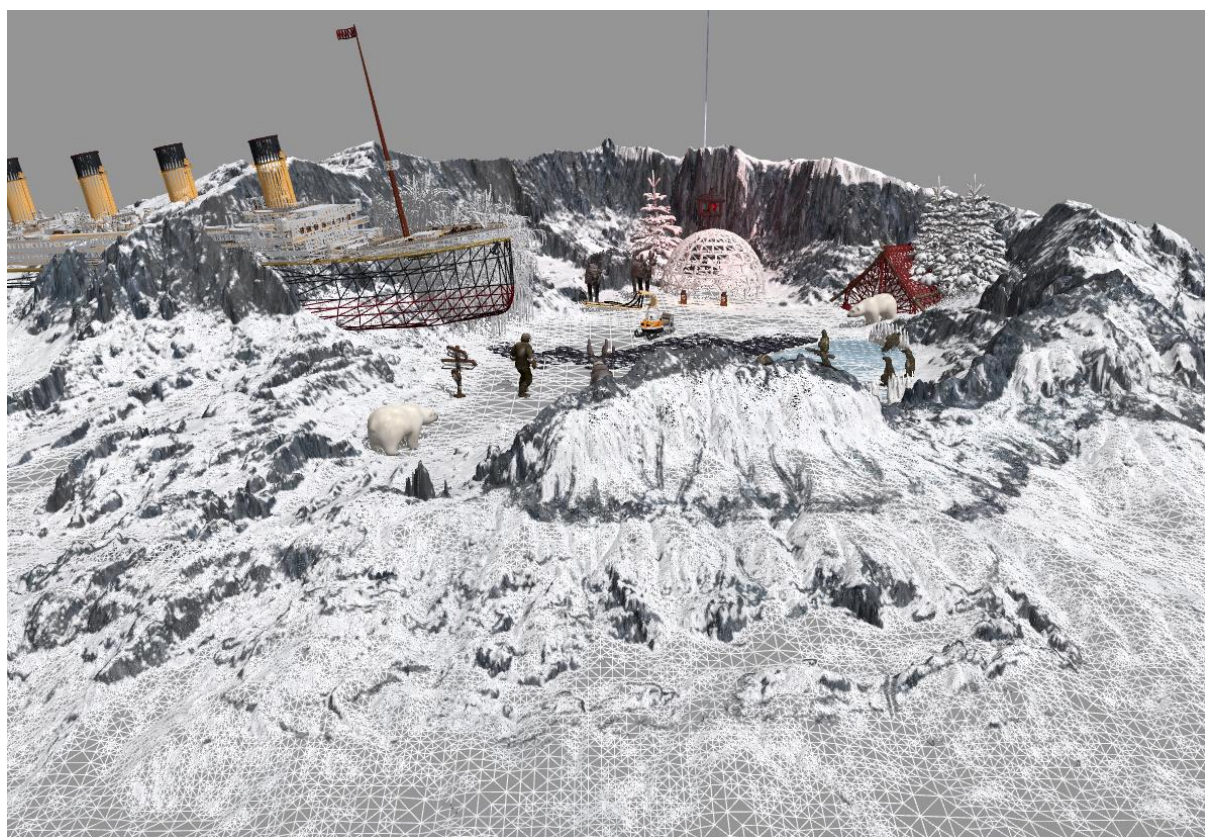


Scena creată

2.2. Funcționalități

Utilizatorul dispune de opțiunea de a interacționa cu scena folosind tastatura și mouse-ul pentru a explora detaliile texturilor obiectelor și diverse perspective ale cadrului. Navigarea în scenă este simplificată prin ajustarea punctului țintă al camerei cu ajutorul mouse-ului, iar tastatura oferă facilități pentru deplasarea în stânga, față, spate sau dreapta, precum și pentru rotirea camerei în jurul originii. Pentru o vizualizare mai detaliată și clară, utilizatorul poate mări fereastra. Controlul rotației luminii este posibil prin intermediul a două taste dedicate.

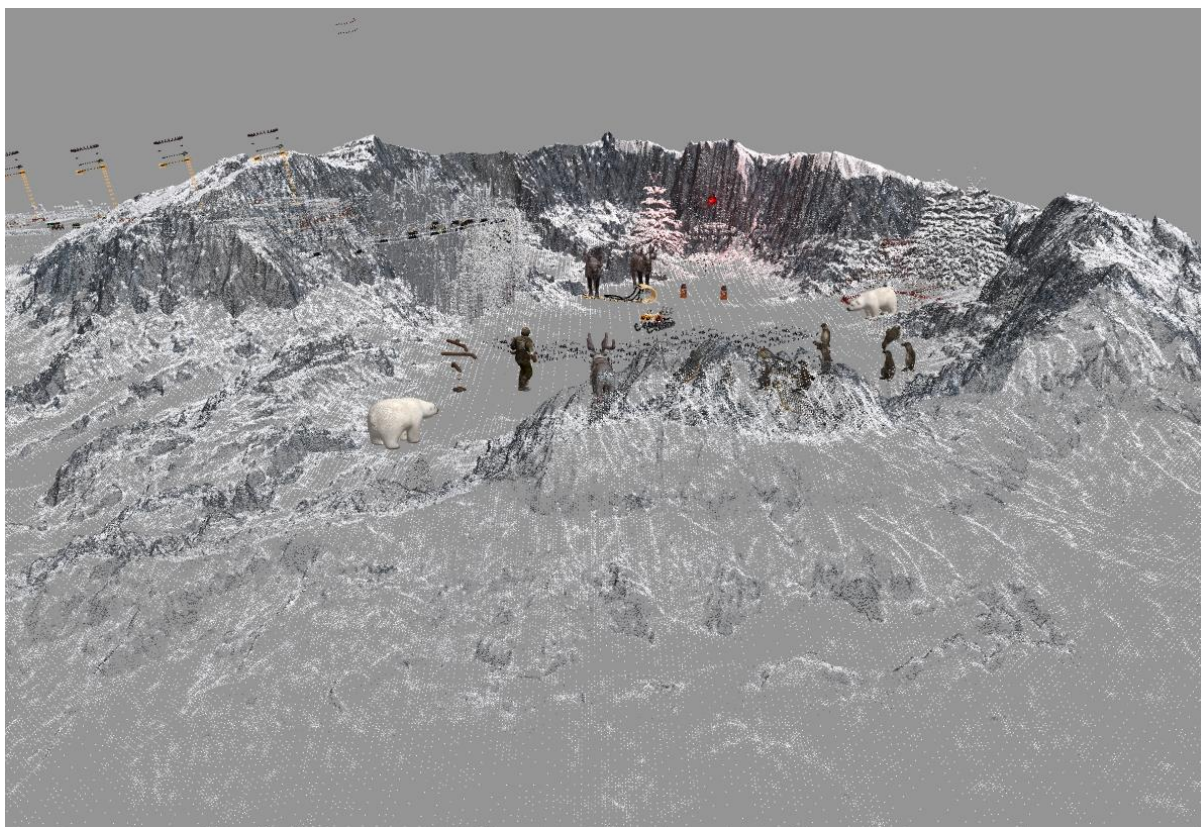
Modurile de afișare solid (Se referă la afișarea unui obiect cu suprafețe pline, adică cu toate fețele sale complet umplute cu culoare sau textură.), wireframe (Este un mod de afișare grafică în care obiectele sunt reprezentate doar printr-un schelet de linii, fără a avea suprafețe pline.) și poligonal (Obiectele nu sunt umplute cu texturi sau culori, ci sunt reprezentate printr-o rețea de poligoane care evidențiază structura lor geometrică.) sunt disponibile.



Modul de vizualizare wireframe

Funcționalitățile adiționale includ vizualizarea hărții de adâncime, activarea ninsorii, ajustarea numărului de fulgi de nea și animația diferitelor obiecte.

Prin intermediul animațiilor de prezentare și a mișcării în scenă, utilizatorul poate observa dinamica umbrelor generate de obiecte în funcție de poziția sursei de lumină. Sursele de lumină punctiforme din scenă contribuie la consolidarea unei impresii realiste în cadrul prezentării complete a scenei 3D.



Modul de vizualizarea poligonal

3. Detalii de implementare

3.1. Funcții și algoritmi

3.1.1. Soluții posibile

Întrucât scena finală trebuia să aibă nuanțe fotorealiste și o anumită complexitate era nevoie de anumiți algoritmi care să ne ajute să realizăm aceste obiective.

- **Efectul de ninsoare:** Am optat pentru implementarea acestui efect prin desenarea de particule (stropi de zăpadă) pe întreaga scenă. Aceste particule sunt generate aleatoriu în spațiul 3D și se deplasează către partea de jos, simulând căderile de zăpadă. Utilizarea `std::random_device` asigură informații aleatoare de înaltă calitate pentru a inițializa generatorul `std::mt19937`. Acesta generează secvențe de numere aparent aleatoare, dar

deterministe, contribuind astfel la obținerea unor numere bine distribuite în aplicație.

- **Tehnica Shadow Mapping:** Adaugă realism modelului de iluminare, folosind texture de adâncime pentru a decide dacă un punct se află în umbră sau nu, privind din perspectiva sursei de lumină. Implementarea algoritmului implică două etape: observarea scenei din punctul de vedere al sursei de lumină și salvarea informațiilor de adâncime într-o textură; compararea adâncimile obiectelor din această perspectivă cu cele salvate în harta de adâncime. Dacă adâncimea obiectului văzut este mai mare decât adâncimea corespunzătoare din harta de adâncime, punctul este în umbră; în caz contrar, este iluminat.



Harta de adâncime

- **Iluminarea:** Am adoptat modelul de iluminare Blinn-Phong, o variantă îmbunătățită a modelului Phong, focalizată în special pe reflexiile speculare pentru suprafețele cu un coeficient scăzut de strălucire. Implementarea include calculul componentelor de iluminare (ambientală, difuză, speculară) adaptate pentru fiecare fragment al scenei. De asemenea, am implementat tehnica de

hărți de umbră pentru a simula umbre realiste în funcție de perspectiva sursei de lumină.

- **Animația pinguinului:** În cadrul funcției `floatingPenguinAnimation()`, timpul (`currentTimeStamp`) este esențial pentru generarea unei animații bazate pe evoluția timpului. Algoritmul se bazează pe momentul actual al aplicației în secunde, actualizarea diferenței de timp între două stări succesive, și calculul unghiului de rotație. Prin actualizarea unghiului la fiecare cadru, se creează iluzia unei mișcări continue a pinguinului. Astfel, se asigură că animația nu depinde direct de viteza de procesare a sistemului, menținând un flux constant indiferent de ritmul de execuție al programului.

3.1.2. Motivarea abordării alese

Alegerea abordării pentru implementarea proiectului s-a bazat pe mai multe considerente fundamentale. În primul rând, selecția tehnicilor și algoritmilor a fost ghidată de obiectivele proiectului de a crea o scenă cu nuanțe fotorealiste și de a integra aspecte complexe în iluminare și animație.

Am considerat că abordarea cea mai adecvată pentru realizarea proiectului implică aplicarea tehnicilor dobândite în timpul laboratoarelor de prelucrare grafică. Materialele corespunzătoare au fost suficient de explicite pentru a oferi sprijinul și înțelegerea necesare în elaborarea proiectului. Prin urmare, am integrat ambele surse de lumină posibile, direcțională și punctiformă, dar și modelul de iluminare Blinn-Phong și tehnica Shadow Mapping.

3.2. Modelul grafic

Scena a fost creată folosind software-ul Blender, mediu în care au fost adăugate pe rând modele 3D găsite pe Internet. Pentru a folosi obiectele modelate, a fost necesară exportarea lor cu extensia `.obj`.

3.3. Structuri de date

Acestea includ:

- Matrice de transformare: Matrice pentru transformări de modele, vizualizare, proiecție și lumină.
- Shader-e: Obiectele folosite în procesul de rasterizare sunt gestionate printr-o structură de tip shader, unde fiecare shader are propriile sale obiecte asociate.
- ID-uri framebuffer: ID-uri pentru framebuffer și textura de adâncime folosite în tehnica de umbrire.
- Skybox: Obiecte pentru gestionarea texturilor skybox-ului - cub cu texturi diferite aplicate pe fiecare față, prin care se afișează o imagine "de fundal" reprezentând partea din scena 3D pe care nu am modelat-o.

Am utilizat preponderent structurile de date furnizate de biblioteca GLM (OpenGL Mathematics). Printre acestea, se evidențiază:

mat<n> și vec<n> din GLM, fiind esențiale pentru manipularea și gestionarea matricelor și vectorilor în contextul grafic.

3.4. Ierarhia de clase

Structura ierarhică a proiectului este organizată în jurul mai multor clase, fiecare îndeplinind funcții specifice în cadrul aplicației:

- Camera.cpp & Camera.hpp gestionează camera în spațiul virtual. Implementează funcționalitățile asociate cu vizualizarea și manipularea perspectivei camerei.
- Mesh.cpp & Mesh.hpp manipulează mesh-urile 3D, inclusiv coordonatele verticiale, normalele și texturile și oferă funcționalități pentru desenarea obiectelor 3D utilizând OpenGL, gestionând buffer-ele de date și interacționând cu shader-ele.

- Model3D.cpp & Model3D.hpp gestionează coordonatele obiectului cu formatul obj și le furnizează unei structuri de date esențiale pentru o reprezentare adecvată a modelelor.
- Shader.cpp & Shader.hpp implementează funcționalitățile legate de gestionarea shaderelor în procesul de randare. Asigurarea comunicării eficiente între elementele vizuale ale proiectului.
- Stb_image.cpp & stb_image.h manipulează imaginile utilizate.
- Tiny_obj_loader.cpp & tiny_obj_loader.h furnizează funcționalitățile necesare pentru încărcarea și manipularea obiectelor 3D în formatul obj.
- Window.cpp & Window.h gestionează funcționalitățile legate de fereastra de vizualizare în cadrul aplicației.
- Main.cpp: Clasa principală a proiectului care coordonează funcțiile și inițializează toate elementele necesare pentru funcționarea aplicației.

Această ierarhie bine structurată facilitează dezvoltarea, întreținerea și extinderea proiectului, oferind o abordare modulară pentru fiecare componentă a aplicației grafice.

În directorul "objects", unde fiecare obiect modelat exportat din Blender este structurat într-un folder separat, include următoarele:

- Scena_statică: Include scena exportată din Blender, împreună cu obiectul, fișierele mtl și imaginile de texturare corespunzătoare.
- Lampă, snowmobil, pinguin, turn cu flacără, fulg de zăpadă: Include obiectele aferente care au fost introduse în scenă prin transformări, pentru a le putea adăuga diferite funcționalități.
- Quad: Reprezintă un dreptunghi în spațiul 3D, desenat pentru a afișa textura generată în timpul prezentării hărții de adâncime.

4. Prezentarea interfeței grafice utilizator / manual de utilizare

Manual de utilizare:

1. În momentul lansării executabilului, după o scurtă perioadă de așteptare, fereastra cu scena se va deschide. În acest moment, animația de prezentare începe automat.

2. Pentru a naviga în scenă și a vizualiza funcționalitățile proiectului, se vor apăsa următoarele:

- ÷ Tasta W – mișcare în față
- ÷ Tasta S – mișcare în spate
- ÷ Tasta A – mișcare la stânga
- ÷ Tasta D – mișcare la dreapta
- ÷ Tasta L – rotire lumină dreapta
- ÷ Tasta J – rotire lumină stânga
- ÷ Tasta Z – vizualizare în modul solid
- ÷ Tasta X – vizualizare în modul wireframe
- ÷ Tasta C – vizualizare în modul poligonal
- ÷ Tasta R – pornirea/oprirea efectului de ninsoare
- ÷ Tasta N – mărirea intensității ninsorii
- ÷ Tasta B – scăderea intensității ninsorii
- ÷ Tasta M – vizualizarea hărții de adâncime
- ÷ Tasta P – vizualizarea scenei pe timp de noapte/zi

Animația snowmobil-ului se poate observa folosind:

- ÷ Tasta I – mișcare în spate
- ÷ Tasta K – mișcare în față

3. Pentru vizualizare full screen se va apăsa iconița specifică.

4. La final, se închide executabilul.



Vizualizarea scenei pe timp ne noapte

5. Concluzii și dezvoltări ulterioare

Flexibilitatea cerințelor a avut un impact fundamental în modelarea proiectului ca o experiență personalizată și satisfăcătoare. Această abordare a stimulat creativitatea și viziunea personală în întregul proces de dezvoltare.

Implicarea în acest proiect a avut o influență semnificativă asupra modului personal de utilizare a software-ului Blender. Am dobândit competențe în manipularea modelelor 3D, texturarea, iluminarea și animația. Procesul creativ din spatele proiectului nu s-a limitat la aspectele tehnice, am realizat că detinerea unei viziuni artistice clare și dezvoltarea abilităților de a manipula detalii sunt la fel de importante pentru crearea unor scene realiste și atrăgătoare.

În ansamblu, acest proiect nu doar că a contribuit la dezvoltarea competențelor tehnice, ci a fost și o experiență formativă care a adus îmbunătățiri semnificative abilităților artistice.

Dezvoltarea continuă a proiectului se concentrează pe extinderea scenei și integrarea unui număr mai mare de obiecte, cu un accent deosebit pe calitate și realism.

Integrarea detecției coliziunilor reprezintă o evoluție ulterioară a proiectului. Această funcționalitate aduce beneficii semnificative în ceea ce privește realismul și coerența scenei. Atunci când două entități interacționează, sistemul poate detecta automat acest eveniment și poate declanșa acțiuni adecvate, precum oprirea mișcării, schimbarea direcției sau activarea unor animații specifice.

Extinderea animației animalelor din cadrul scenei poate fi considerată o modalitate de dezvoltare suplimentară. Adăugarea unor mișcări fluide și naturale ale membrilor animalelor poate contribui la crearea unei atmosfere mai vii și autentice în cadrul virtual.

6. Referințe

Pentru obținerea obiectelor și texturilor utilizate în cadrul proiectului, am explorat și folosit resurse disponibile pe Internet. Acestea au fost obținute de pe patru pagini web specializate care oferă modele 3D gratuite. Sursele principale de unde au fost descărcate, includ:

- <https://www.turbosquid.com/>
- <https://sketchfab.com/>
- <https://www.cgtrader.com/>
- <https://free3d.com/>
- <https://moodle.cs.utcluj.ro/>