



**TECHNICAL
UNIVERSITY**
OF CLUJ-NAPOCA
ROMANIA

FACULTATEA: Automatică și Calculatoare
SPECIALIZAREA: Calculatoare și Tehnologia Informației
DISCIPLINA: Prelucrare Grafică
PROIECT: Scena 3D

Îndrumător laborator:
Alexandru Horea Anisorac

Student:
Galiș George-Laurențiu



Cuprins

1. Cerinta

2. Scenariul

3. Detalii de implementare

- **Functii si algoritmi**
- **Modelul grafic**
- **Structuri de date**
- **Ierarhia de clase**

4. Prezentarea interfetei grafice

5. Concluzii și dezvoltări ulterioare

6. Referinte

Cerinta

Proiectele au ca și scop realizarea unei prezentări fotorealiste a unor scene de obiecte 3D utilizând librăriile prezentate la laborator (OpenGL, GLFW, GLM, etc.). Utilizatorul trebuie să aibă posibilitatea de a controla scena prin intermediul mausului și tastaturii.

- **(2p)** vizualizarea scenei: scalare, translație, rotație, mișcarea camerei
 - utilizând tastatura sau mausul
 - utilizând animații de prezentare
- **(1p)** specificarea surselor de lumina (cel puțin două surse de lumină diferite)
- **(0.5p)** vizualizare scenă în modurile solid, wireframe, poligonal și smooth
- **(1p)** maparea texturilor și definirea materialelor
 - calitatea texturilor și nivelul de detaliu al acestora
 - maparea texturilor pe obiecte
- **(1p)** exemplificarea generării umbrelor
- **(0.5p)** exemplificarea animării diferitelor componente ale obiectelor
- **(3p)** fotorealism, complexitatea scenei, nivelul de detaliere al modelării, dezvoltarea diferiților algoritmi și implementarea acestora (generare dinamică de obiecte, detecția coliziunilor, generarea umbrelor, ceață, ploaie, vânt), calitatea animațiilor, utilizarea diferitelor surse de lumină (globală, locală, de tip spot)
- **(1p)** documentația (obligatorie)



Scenarii

La rularea programului, utilizatorul poate vedea o fereastră deschisă după încărcarea tuturor obiectelor și texturilor (proces ce durează câteva secunde), o fereastră ce conține o lume 3d cu obiecte.

Scena conține următoarele obiecte:

- pamantul - care reprezintă partea verde din scena și pe care se află celelalte obiecte. Acesta conține și denivelări
- lacul - care se află în centrul scenei și care “izvoraste” de pe un deal
- două case abandonate
- o mașină abandonată
- două fantani
- cai
- moara
- un pod rupt
- brazi
- drum de țară

Scena poate fi traversată cu ajutorul mouse-ului și tastelor w,a,s,d, utilizatorul putând să se plimbe oriunde în scena

Mai mult de atât, se poate alege unul din modurile de prezentare cu ajutorul tastelor 1,2,3:

- Solid
- Wireframe
- Point



Detalii de implementare

Funcții și algoritmi

Principalele funcții care realizează proiectarea acestor obiecte într-o scenă sunt:

- `renderScene()` care desenează scena principală cu obiectele acestora,
- `initUniforms` care generează matricele model, view și projection pentru obiectele din scenă,
- `initShaders()` care initializează fișierele `shader.frag` și `shader.vert`,
- `initModels()` care initializează fișierele `.obj` în scenă
- `processMovement` care citește tastele apasate și generează o acțiune
- `mouseCallBack` care face posibilă mișcarea în scenă folosind mouse-ul
- `main()` care înglobează toate aceste funcții

Unul dintre algoritmi folosiți este algoritmul de iluminare Phong care reflectă lumina de pe suprafețele obiectelor aflate în scenă.



Modelul grafic

Modelele grafice utilizate sunt cele prezentate in laboratoarele parcurse pe parcursul semestrului, mai exact modelul de iluminare Bill-Phong si modelul ShadowMapping care foloseste tipurile de lumini difuze, ambientale si speculara

Structuri de date

Tipurile de date si clasele folosite sunt cele din biblioteca GLM cum ar fi:

- mat4
- vec3
- Camera
- Shader
- Model3D
- Window

Ierarhia de clase

Clasele care a facut dezvoltarea aplicatiei mai usoara sunt:

- Camera.hpp: controlul camerei
- Mesh.hpp: defineste varfurile unui obiect
- Model3D.hpp: creeaza un nou model 3D
- Shader.hpp: incarca un shader
- glm.hpp – biblioteca pentru calculul matematic
- GLEW.h si GLFW.h – pentru functionalitate/randari

Functii speciale

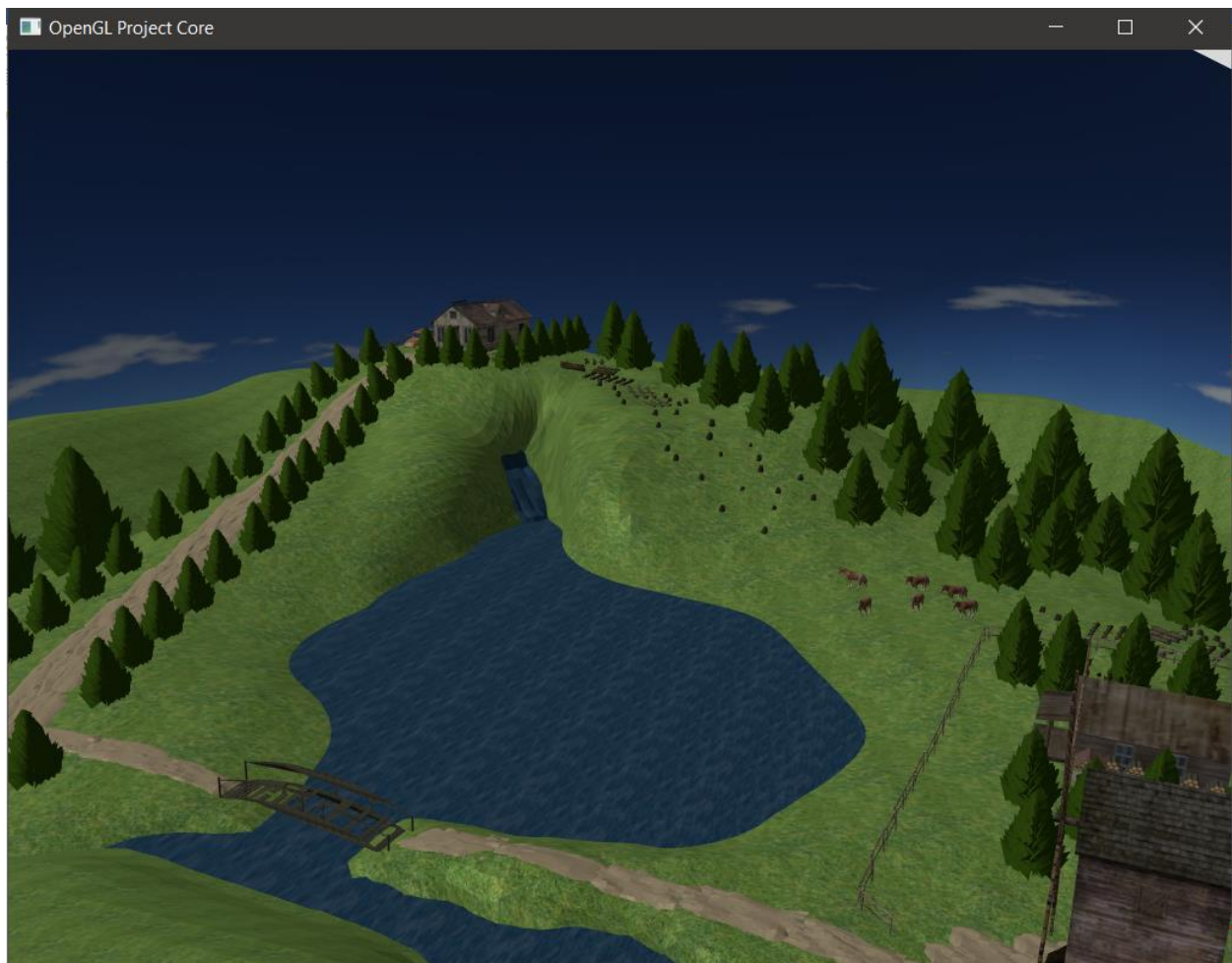
Clasele care implementeaza niste actiuni speciale sunt:

- `PlaySound(TEXT("eagle_sound.wav"), NULL, SND_ASYNC | SND_FILENAME | SND_LOOP);` - genereaza un sunet de fundal(nefunctional)
- `Void mouseCallback(GLFWwindow* window, double xpos, double ypos)`- miscarea camerei cu ajutorul mouse-lui

Prezentarea interfetei grafice utilizator

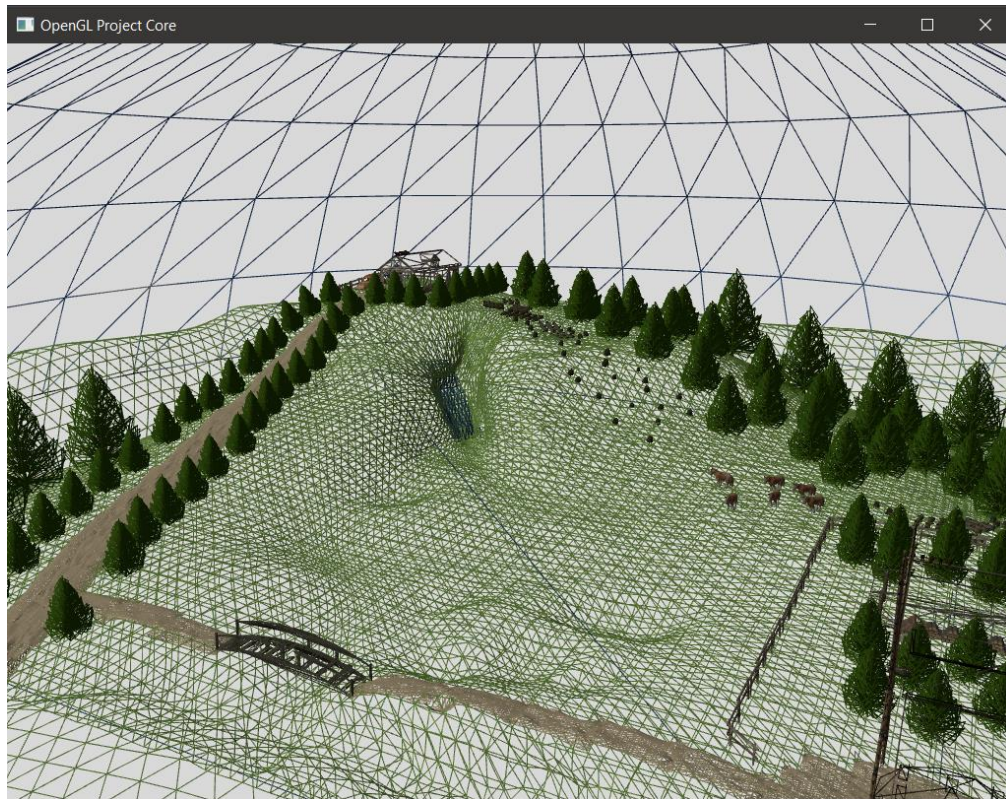
Mai jos se vor prezenta scena cu toate modelele folosite si modelele separate.

Scena solid

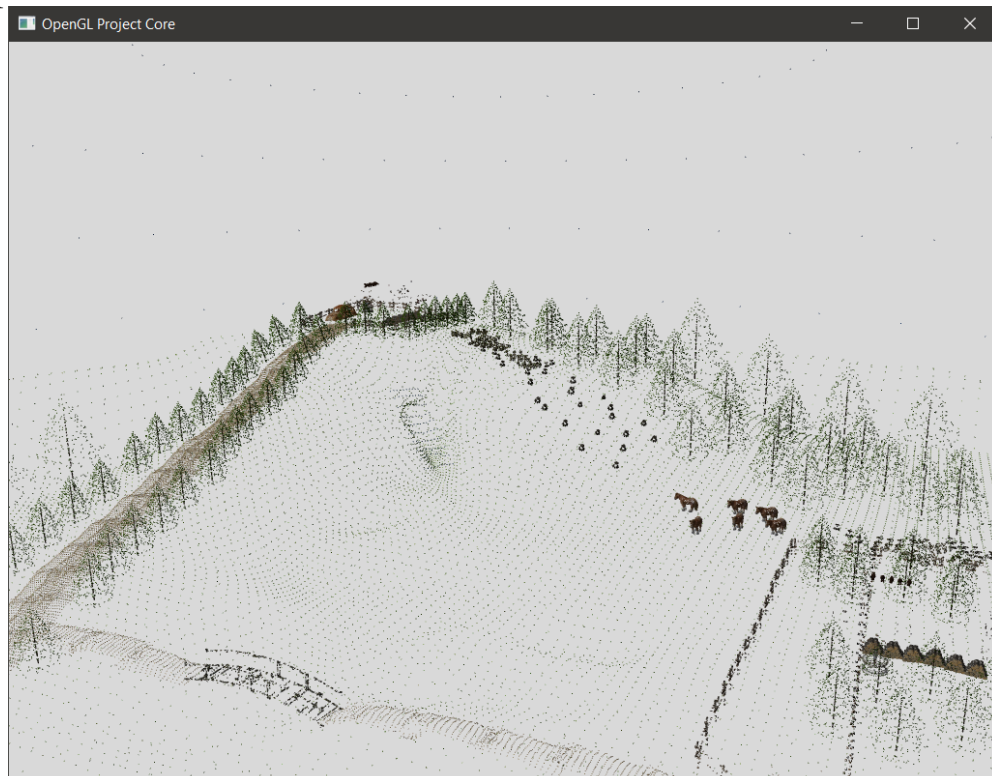




Scena line



Scena point





Moara



Taste folosite pentru generarea actiunilor

- “w” – miscare in fata
- “s” – miscare in spate
- “a” – miscare in stanga
- “d” – miscare in dreapta
- “q” – rotire scena in stanga
- “e” – rotire scena in dreapta
- “mouse” – rotatie camera
- “z” – generare sunet(nefunctional)
- “k” – rotrire sursa de lumina la stanga
- “l” – rotire sursa de lumina la dreapta

Concluzi si dezvoltari ulterioare

În concluzie, crearea unei astfel de aplicații duce la înțelegerea mult mai ușoară a unei interfete grafice cu nenumărate tipuri de obiecte și cu texturi. Astfel se poate vizualiza mult mai ușor dezvoltarea unor scene cu obiecte dinamice și cu mișcări de cameră ca de exemplu jocurile video. Aceste scene pot fi create și modelate în diferite programe de modelare 3D cum ar fi Blender.

Pentru a putea îmbunătăți acestei aplicații se poate reduce numărul de forme aflate pe obiecte cum ar fi copacii folosind diferite iluzii optice (un copac poate fi reprezentat doar cu două planuri aflate la 90 de grade, nu cu mai multe). Acest lucru duce la reducerea memoriei. Mai mult de atât se pot adăuga diferite efecte asupra obiectelor cum ar fi coliziuni, generare de umbre, creșterea rezoluției texturilor aflate pe obiect.

Referinte

<https://learnopengl.com/>

<http://www.free3d.com/>

<https://www.youtube.com/>

<https://moodle.cs.utcluj.ro/course/view.php?id=304>