

# **Documentație Proiect Prelucrare Grafica**

Toth Antonio-Roberto, grupa 30238

28.12.2025

# Cuprins

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

# 1 Prezentarea temei

Scopul acestui proiect este realizarea unei aplicații grafice 3D interactive, utilizând biblioteca grafică OpenGL (Open Graphics Library). Proiectul urmărește simularea unei scene fotorealiste de iarnă, punând accent pe tehniciile moderne de randare, iluminare și gestionare a resurselor grafice.

Obiectivul principal a fost crearea unui mediu virtual complex, compus din multiple obiecte 3D, care să permită utilizatorului explorarea scenei în timp real. Aplicația demonstrează utilizarea shaderelor programabile (GLSL) pentru a implementa efecte avansate precum umbre dinamice (Shadow Mapping), iluminare Blinn-Phong, ceată și un sistem de particule pentru simularea zăpezii [1].

Proiectul integrează biblioteci auxiliare standard pentru dezvoltarea grafică: **GLFW** pentru gestionarea ferestrei și a contextului, **GLEW** pentru încărcarea extensiilor OpenGL și **GLM** pentru operații matematice vectoriale și matriceale.

## 2 Scenariul

### 2.1 Descrierea scenei și a obiectelor

Scena este plasată într-un peisaj de iarnă, construit pe un teren plan acoperit cu zăpadă. Elementele constitutive ale scenei au fost alese pentru a crea o atmosferă rustică și rece:

- **Elemente naturale:** O pădure de conifere (brazi) de diverse dimensiuni, teren texturat cu zăpadă.
- **Structuri antropice:** O casă de lemn, un turn de veghe (watch tower), o moară de vânt funcțională, un gard și o fântână.
- **Surse de lumină și decor:** Un foc de tabără (campfire) animat, o lanternă atașată unei structuri, un urs.
- **Fundal:** Un Skybox care completează orizontul cu imagini specifice unui peisaj montan.

### 2.2 Funcționalități

Aplicația oferă utilizatorului un set extins de funcționalități pentru interacțiune și control:

- **Navigare:** Controlul camerei de tip *Fly-Camera* folosind tastatura (WASD) și mouse-ul.
- **Tur Automat:** Posibilitatea de a activa o animație predefinită a camerei care orbitează scena (Tasta 'P').
- **Iluminare Dinamică:** Controlul surselor de lumină: Soare (Tasta 'K'), Lanternă (Tasta 'L'), Foc de tabără (Tasta 'C').
- **Condiții Meteo:** Activarea/Dezactivarea căii (Tasta 'F') și a ninsorii (Tasta 'N').
- **Moduri de Randare:** Vizualizarea scenei în modurile Solid, Wireframe și Point (Tastele 1, 2, 3).

- **Coliziuni:** Implementarea coliziunii simple cu solul, împiedicând camera să treacă sub nivelul terenului.

## 3 Detalii de implementare

### 3.1 Funcții și algoritmi

#### 3.1.1 Soluții posibile

Pentru realizarea umbrelor, au fost analizate două metode principale:

1. **Shadow Volumes:** O tehnică bazată pe geometrie care generează umbre precise (hard shadows), dar care este costisitoare computațional pentru obiecte cu număr mare de poligoane.
2. **Shadow Mapping:** O tehnică bazată pe texturi (image-space), utilizată pe scară largă datorită eficienței sale și posibilității de a crea umbre mai naturale (soft shadows).

Pentru sistemul de particule (ninsoare), s-a considerat utilizarea *Geometry Shaders* pentru generarea fulgilor sau actualizarea pozițiilor pe CPU.

#### 3.1.2 Motivarea abordării alese

Am ales implementarea algoritmului de **Shadow Mapping** [2]. Motivul principal este performanța superioară în scene complexe și ușurința integrării în pipeline-ul de shadere existent. Umbrele sunt generate în doi pași:

1. Randarea scenei din perspectiva luminii într-un *Framebuffer Object* (FBO) pentru a obține harta de adâncime (Depth Map).
2. Randarea finală, unde se compară adâncimea fragmentului curent cu valoarea din Depth Map.

Pentru ninsoare, am ales actualizarea pozițiilor pe CPU și randarea folosind **GL\_POINTS**. Deoarece numărul de particule este limitat (50.000), actualizarea pe CPU este suficient de rapidă și permite o logică flexibilă de "respawn" a fulgilor când ating solul.

### 3.2 Modelul grafic

Modelul grafic este bazat pe pipeline-ul programabil OpenGL.

- **Vertex Shader:** Realizează transformările de coordonate (Model-View-Projection), calculează vectorii normali și coordonatele în spațiul luminii pentru umbre.
- **Fragment Shader:** Implementează modelul de iluminare Blinn-Phong. Aceasta combină componenta ambientală, difuză și speculară. De asemenea, aplică factorul de umbră (Shadow calculation) și mixează culoarea finală cu culoarea cerii (Fog) în funcție de distanța față de cameră.

### 3.3 Structuri de date

În cadrul aplicației au fost definite structuri specifice pentru gestionarea eficientă a datelor:

```
1 struct SnowParticle {  
2     glm::vec3 position;  
3     glm::vec3 velocity;  
4     float size;  
5     float lifetime;  
6 };
```

Vectorul `std::vector<SnowParticle> snowParticles` gestionează toți fulgii de zăpadă, permitând iterarea rapidă pentru actualizarea pozițiilor în fiecare cadru. Pentru obiectele 3D, se utilizează clasa `gps::Model3D` care încapsulează mesh-urile și texturile.

### 3.4 Ierarhia de clase

Proiectul este structurat modular, utilizând clase helper pentru a abstractiza complexitatea OpenGL:

- **Window:** Gestionează crearea ferestrei și contextul OpenGL via GLFW.
- **Camera:** Gestionează matricea de vizualizare (*View Matrix*), poziția și vectorii de orientare (Front, Right, Up). Include metoda `rotate(pitch, yaw)` pentru controlul mouse-ului.
- **Shader:** Se ocupă de citirea, compilarea și link-area shaderelor Vertex și Fragment.
- **Model3D / Mesh:** Gestionează încărcarea fișierelor .obj și randarea geometriei folosind VAO/VBO.
- **SkyBox:** O clasă specializată pentru încărcarea și afișarea Cubemap-ului.

## 4 Prezentarea interfeței grafice utilizator / Manual de utilizare

Interfața aplicației este vizuală, controlul realizându-se prin periferice standard. La porțire, utilizatorul este plasat în centrul scenei.

**Tabel comenzi:**

Tastă	Funcționalitate
W, A, S, D	Deplasare cameră (Față, Stânga, Spate, Dreapta)
Mouse	Orientare privire (Rotire cameră)
P	Start/Stop Tur Automat (Prezentare cinematică)
F	Activare/Dezactivare Ceată (Fog)
N	Activare/Dezactivare Ninsoare
K	Activare/Dezactivare Lumină Soare
L	Activare/Dezactivare Lanterna (Point Light)
C	Activare/Dezactivare Foc de tabără (Flickering Light)
1, 2, 3	Moduri randare: Solid, Wireframe, Point
ESC	Ieșire aplicație



Figura 1: Captură de ecran din aplicație - Modul Solid cu umbre activate.



Figura 2: Captură de ecran din aplicație - Modul de ceață activat.

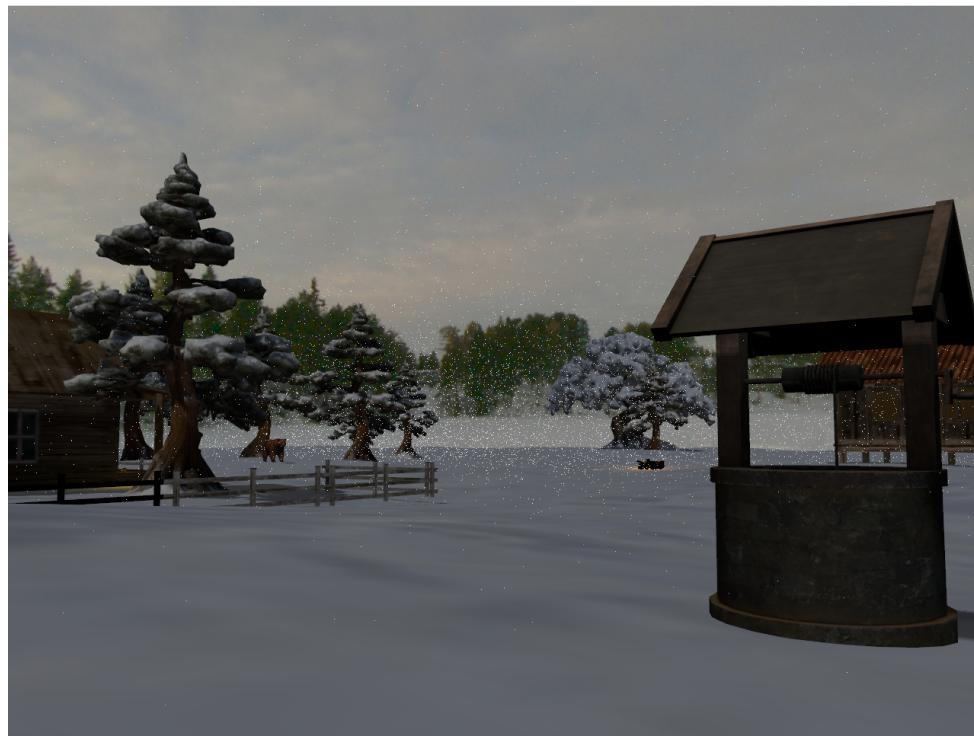


Figura 3: Captură de ecran din aplicație - Modul de ninsoare activat,

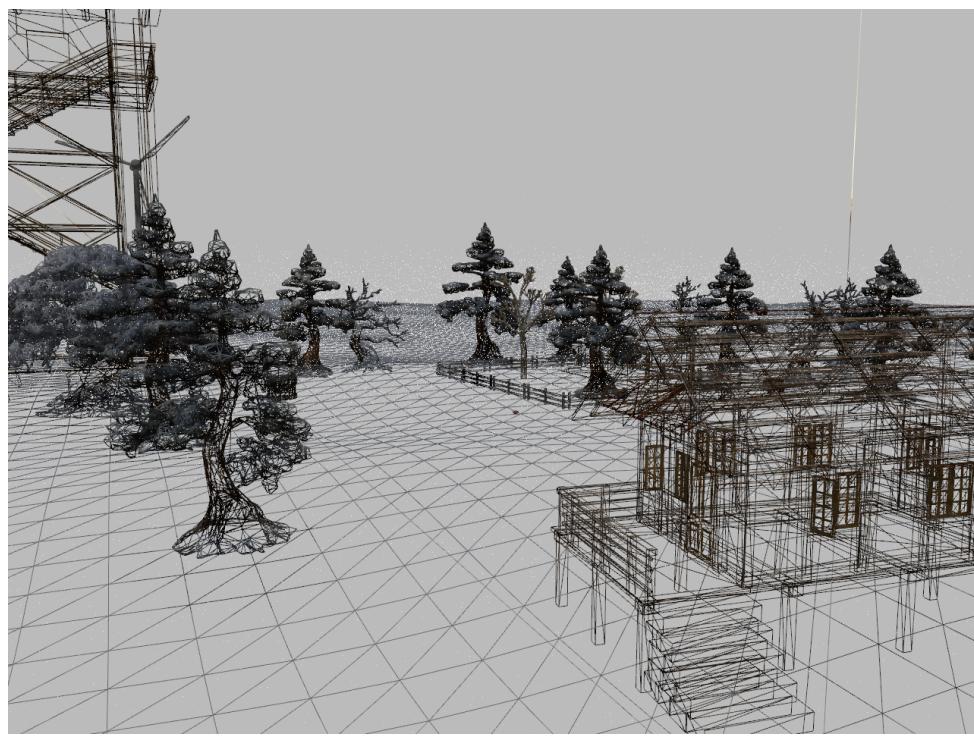


Figura 4: Captură de ecran din aplicație - Modul wireframe activat.

## 5 Concluzii și dezvoltări ulterioare

Proiectul a reușit implementarea cu succes a unei scene 3D complexe, îndeplinind cerințele de fotorealism prin utilizarea tehnicii avansate de iluminare și texturare. Implementarea umbrelor dinamice și a sistemului de particule contribuie semnificativ la imersiunea vizuală.

**Dezvoltări ulterioare:** Pe viitor, proiectul poate fi extins prin:

- Adăugarea unui sistem de coliziuni complex (bounding box) pentru toate obiectele din scenă, nu doar pentru sol.
- Implementarea sunetului 3D (ex: sunetul focului de tabără care devine mai puternic pe măsură ce te apropii).
- Utilizarea *Normal Mapping* pentru a adăuga detalii de relief pe suprafetele obiectelor fără a crește numărul de poligoane.

## 6 Referințe

### Bibliografie

- [1] Joey de Vries, *Learn OpenGL - Extensive Tutorial Resource*, <https://learnopengl.com/>,
- [2] OpenGL Tutorial, *Tutorial 16: Shadow Mapping*, <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-16-shadow-mapping/>.
- [3] Laborator Grafică pe Calculator, *UTCN*, Resurse interne curs.
- [4] The OpenGL Extension Wrangler Library (GLEW), <http://glew.sourceforge.net/>.