**Universitatea Tehnică din Cluj-Napoca**

**Catedra de Calculatoare**

**Prelucrare Grafică**

**Documentația proiectului**

**Student:** Racz Milán

**Grupa:** 30236

**Data:** 12. 01. 2022

1. **Cuprins**

[**2.** **Prezentarea temei** 3](#_Toc92924027)

[**3.** **Scenariul** 3](#_Toc92924028)

[**3.1** **Descrierea scenei și a obiectelor** 3](#_Toc92924029)

[**3.2** **Funcționalități** 3](#_Toc92924030)

[**4.** **Detalii de implementare** 3](#_Toc92924031)

[**4.1.** **Funcții și algoritmi** 3](#_Toc92924032)

[**4.2.** **Modelul grafic** 4](#_Toc92924033)

[**4.3.** **Structuri de date** 4](#_Toc92924034)

[**4.4.** **Ierarhia de clase** 4](#_Toc92924035)

[**5.** **Prezentarea interfeței grafice utilizator / manual de utilizare** 4](#_Toc92924036)

[**6.** **Concluzii și dezvoltări ulterioare** 5](#_Toc92924037)

[**7.** **Referințe** 5](#_Toc92924038)

# **Prezentarea temei**

Scopul acestui proiect a fost crearea unei scene fotorealistice, compuse din diferite obiecte 3D, utilizând biblioteca OpenGL. Utilizatorul poate să interacționeze cu scena cu ajutorul tastaturii și al mouse-ului.

# **Scenariul**

## **Descrierea scenei și a obiectelor**

Scena este compusă dintr-o casă, o mașină, o moară, o lampă, trei copaci în jurul casei și un gard în jurul casei. Casa se află în mijlocul scenei pe un plan cu o textură seamless de iarbă. Mașina se află pe un drum în fața casei, parcată, și este compusă din trei materiale. Una din aceste materiale este sticlă, care este transparentă. Moara se află lângă casă și roata ei se învârte. Lampa este montată pe un stâlp al casei și lumina punctiformă este pe coordonatele acesteia.

## **Funcționalități**

Utilizatorul se poate deplasa prin scenă folosind tastele W, A, S, D, respectiv poate modifica direcția de vizualizare a camerei prin mișcarea mouse-ului. Lumina direcțională poate fi deplasată în jurul scenei cu ajutorul tastelor J și L, și umbrele se mișcă împreună cu aceasta. Lumina punctiformă poate fi aprinsă sau stinsă prin apăsarea tastei C. Apăsând tasta Z camera sare la o coordonată de start și începe să se miște în jurul scenei, ajungând înapoi în jurul punctului de start. Scena se poate roti cu tastele Q și E. Cu tasta T se poate vizualiza scena în modul wireframe. Prin apăsarea tastei Y se poate schimba timpul zilei între zi și noapte.

# **Detalii de implementare**

## **Funcții și algoritmi**

Există mai multe funcții în main.cpp pentru inițializarea proiectului, de exemplu initOpenGLWindow(), setWindowCallbacks(), initOpenGLState(). În funcția initModels() se încarcă obiectele 3D și skybox-ul. În initShaders() sunt încărcate shaderele cu fișierele cu extensie .vert și .frag. Funcțiile processMovement() și processAnimation() sunt folosite pentru a realiza mișcările prin scenă cu ajutorul tastelor, respectiv pentru a realiza animația de prezentare. Funcția initUniforms() inițializează variabilele uniform ale shaderilor. Funcțiile initFBO() și computeLightSpaceTrMatrix() sunt folosite în calculul umbrelor. În funcțiile renderScene() și renderObjects() se realizează transformările obiectelor și ale luminilor, se actualizează variabilele uniform ale shaderelor și se construiește scena cu obiectele 3D și umbrele lor.

În shaderul basic.frag am implementat algoritmii de calcul pentru reflecții, ceață, umbră și lumină punctiformă. Lumina punctiformă este calculată doar când aceasta este aprinsă. Reflecțiile apar când în funcția renderObjects() este specificat că un anumit obiect are suprafață reflectivă (de exemplu iarba sau drumul nu au). Algoritmii au fost implementați cu ajutorul tutorialelor de pe site-ul [1].

## **Modelul grafic**

Obiectele din scenă sunt obiecte 3D, având extensia .obj, care au fost descărcate de pe site-urile: [2], [3], [4]. Scena a fost construită prima dată în Blender, pentru a poziționa texturile și pentru a scala obiectele, ca acestea să aibă dimensiuni realistice. În cazul morii și al mașinii, unele componente ale obiectelor au fost exportate separat (roata morii, pentru a o roti separat, și geamurile mașinii, pentru că aceasta trebuia să fie transparentă).

## **Structuri de date**

Obiectele vec3, mat3 și mat4 din librăria glm sunt folosite ca și structuri de date pentru a stoca, de exemplu, informațiile despre cameră, pozițiile luminilor și poziția nodurilor obiectelor sau fragmentelor.

## **Ierarhia de clase**

Clasele Mesh și Model3D sunt folosite pentru definirea obiectelor și pentru aplicarea texturilor asupra acestora. Clasa Shader ajută la compilarea și utilizarea programelor de shader din aplicație. Clasa Camera este folosită pentru a schimba poziția sau direcția de vizualizare a camerei. Mai există clasele Window și SkyBox care definesc fereastra, respectiv skyboxul. Aceste clase sunt instanțiate și folosite în main.cpp.

# **Prezentarea interfeței grafice utilizator / manual de utilizare**

Utilizatorul poate folosi următoarele taste pentru a efectua diferitele funcționalități din scenă:

* W – deplasarea camerei în față
* S – deplasarea camerei în spate
* A – deplasare camerei în stânga
* D – deplasare camerei în dreapta
* Q – rotirea scenei în jurul originii spre stânga
* E – rotirea scenei în jurul originii spre dreapta
* Z – pornirea animației de prezentare
* J – deplasare luminii direcționale în jurul originii spre dreapta
* L – deplasare luminii direcționale în jurul originii spre stânga
* C – aprinderea sau stingerea luminii punctiforme
* T – vizualizare scenă în modul wireframe
* Y – se poate schimba timpul zilei

# **Concluzii și dezvoltări ulterioare**

În concluzie, am reușit să creez o scenă fotorealistică alcătuită din modele 3D și am implementat majoritatea funcționalităților propuse. Ca și o dezvoltare ulterioară aș putea implementa și alți algoritmi, cum ar fi detecția coliziunilor sau refracțiile. Aș putea să extind scena cu mai multe obiecte 3D și mai multe animații sau funcționalități.

# **Referințe**

[1] <https://learnopengl.com/>

[2] https://free3d.com/

[3] https://www.turbosquid.com/

[4] https://www.cgtrader.com/