



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Scenă 3D

Prelucrare Grafică

Moldovan Maria Alexandra
Grupa 30234

FACULTATEA DE AUTOMATICĂ
și CALCULATOARE

Ianuarie 2024

Cuprins

1	Prezentarea temei	2
1.1	Descrierea scenei și a obiectelor	2
1.2	Functionalități	3
2	Detalii de implementare	3
2.1	Funcții și algoritmi	3
2.2	Ierarhia de clase	5
3	Prezentarea interfeței grafice. Manual de utilizare	5
4	Concluzii și dezvoltări ulterioare	6
5	Referințe	6

1 Prezentarea temei

Tematica acestui proiect are în vedere o așezare rurală, într-un cadru cuprins de dealuri. Scena a fost creată în *Blender*, iar ulterior au fost adăugate diverse efecte pentru a crea iluzia realismului și un nivel înalt de detaliere al modelului.

Scopul proiectelor este realizarea unei prezentări fotorealiste a unor scene de obiecte 3D, utilizând librăriile prezentate în cadrul laboratorului (OpenGL, GLFW, GLM, etc.). Utilizatorul trebuie să aibă posibilitatea de a controla scena prin intermediul mouse-ului și tastaturii.

1.1 Descrierea scenei și a obiectelor

Scena reprezintă o fermă situată într-un peisaj înconjurat de dealuri cu textură de iarbă. Ferma dispune de teren arabil, câteva bânci, un lac în care înoată rațe, și un țarc în care se află cai, vaci și porci. Printre aceste elemente se găsesc și niște elemente de detaliu, obiecte care fac parte din aceeași temă: butoaie de lemn, magazie mică de depozitare, o fântână și un tractor lângă porțiunea de pământ înconjurată de gard. Am ales această idee pentru a recrea elemente similare jocului meu preferat, *Stardew Valley*, în care jucătorul moștenește o fermă, cu mențiunea că acel joc este în format 2D.



Figura 1: Scena în ansamblu

1.2 Funcționalități

Proiectul are ca funcționalități implementate vizualizarea scenei cu modul scalare, translație, rotație și mișcarea camerei, folosind tastatura, ultima fiind posibilă și cu mouse-ul. Sursa de lumină este una globală.

Efectele aplicate asupra acestei scene, prin dezvoltarea diferenților algoritmi și implementarea acestora, sunt umbrele și ceața.



Figura 2: Umbră

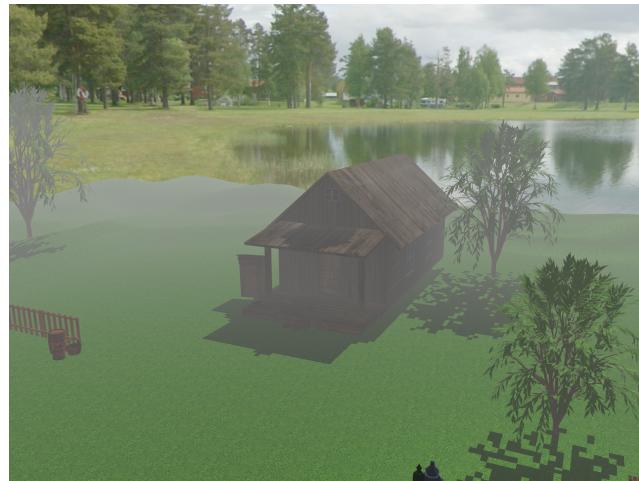


Figura 3: Ceață

2 Detalii de implementare

2.1 Funcții și algoritmi

Pentru a realiza această scenă, a fost nevoie de o funcție care să încarce obiectele în memoria proiectului. Pentru a putea fi vizualizate aceste obiecte încărcate, s-a implementat un algoritm de desenare a acestora, prin care se resetează matricea asupra căreia se vor efectua modificări.

```
1 void initObjects() {  
2     ground.LoadModel("objects/fara_rata/fara_rata.obj");
```

```

3     // scena fara rata care trebe sa se miste
4     rata.LoadModel("objects/rata/rata.obj");
5     // rata care trebe sa se miste
6 }
7
8 // ...
9
10 void drawObjects(gps::Shader shader, bool depthPass) {
11
12     shader.useShaderProgram();
13
14     model = glm::rotate(glm::mat4(1.0f), glm::radians(angleY), glm::vec3(0.0f, 1.0f, 0.0f));
15     glUniformMatrix4fv(glGetUniformLocation(shader.shaderProgram, "model"),
16                         1, GL_FALSE, glm::value_ptr(model));
17
18     if (!depthPass) {
19         normalMatrix = glm::mat3(glm::inverseTranspose(view * model));
20         glUniformMatrix3fv(normalMatrixLoc, 1, GL_FALSE, glm::value_ptr(normalMatrix));
21     }
22
23     rata.Draw(shader);
24
25     model = glm::translate(glm::mat4(1.0f), glm::vec3(0.0f, -1.0f, 0.0f));
26     model = glm::scale(model, glm::vec3(0.5f));
27     glUniformMatrix4fv(glGetUniformLocation(shader.shaderProgram, "model"),
28                         1, GL_FALSE, glm::value_ptr(model));
29
30     if (!depthPass) {
31         normalMatrix = glm::mat3(glm::inverseTranspose(view * model));
32         glUniformMatrix3fv(normalMatrixLoc, 1, GL_FALSE, glm::value_ptr(normalMatrix));
33     }
34
35     ground.Draw(shader);
36 }
```

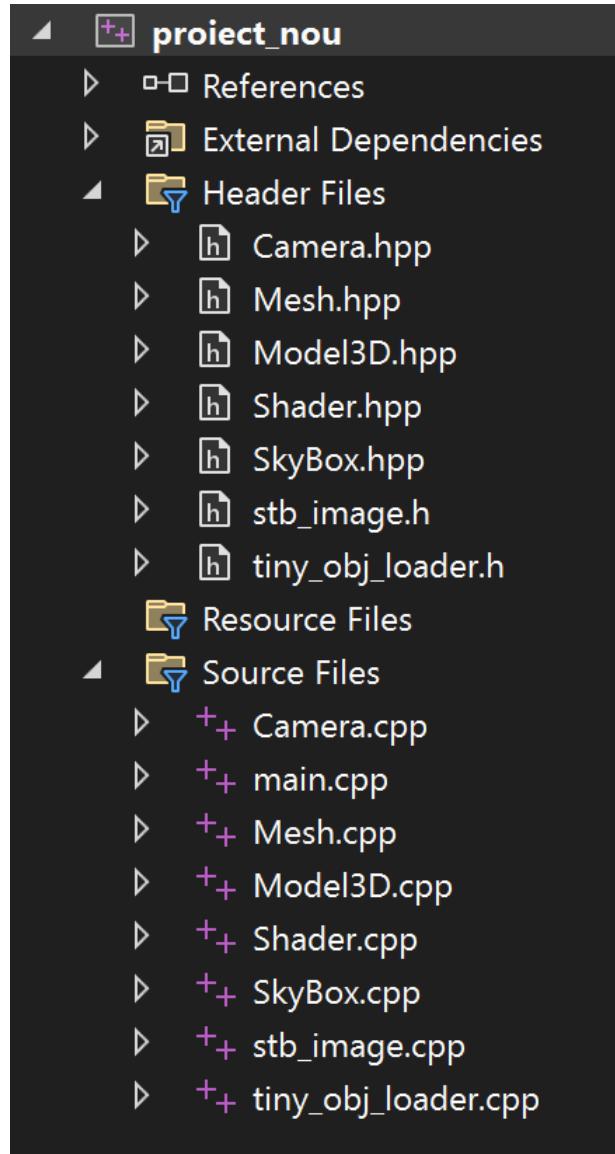
Pentru a anima o rață specifică, am exportat obiectul separat față de întreaga scenă, și asupra lui am aplicat operații de translație și rotație.

```

1 void renderRata(gps::Shader shader) {
2     shader.useShaderProgram();
3
4     float oscillationAmplitude = 2.0f;
5     float oscillationFrequency = 0.025f;
6     float phaseShift = glm::radians(180.0f);
7
8     float oscillationX = oscillationAmplitude *
9         sin(glfwGetTime() * oscillationFrequency * phaseShift);
10
11    glm::mat4 rataModel = glm::rotate(glm::mat4(1.0f), glm::radians(angleY += 0.01f),
12                                     glm::vec3(0.0f, 1.0f, 0.0f)) * glm::translate(glm::mat4(1.0f),
13                                     glm::vec3(oscillationX, 0.0f, 0.0f));
14
15    normalMatrix = glm::mat3(glm::inverseTranspose(view * rataModel));
16    glUniformMatrix3fv(normalMatrixLoc, 1, GL_FALSE, glm::value_ptr(normalMatrix));
17
```

```
18     glUniformMatrix4fv(modelLoc, 1, GL_FALSE, glm::value_ptr(rataModel));  
19 }
```

2.2 Ierarhia de clase



3 Prezentarea interfeței grafice. Manual de utilizare

Utilizatorul poate interacționa cu scena în mai multe moduri:

- Mișcarea direcției camerei – prin mișcarea mouse-ului
- Mișcarea poziției camerei – prin intermediul tastelor W (în față), A (în stânga), S (în spate) și D (în dreapta)
- Modificarea factorului de ceată – prin intermediul tastelor F (mai mult) sau G (mai puțin)
- Închiderea aplicației se poate realiza prin apăsarea tastei ESC (Escape).

4 Concluzii și dezvoltări ulterioare

Proiectul m-a familiarizat cu procesul de asamblare a unei scene 3D, oferindu-mi oportunitatea de a dezvolta competențe semnificative în acest domeniu. Am experimentat cu diferite elemente și am învățat să creez o scenă tridimensională captivantă.

Pentru a îmbunătăți proiectul, intenționez să adaug animații suplimentare pentru a permite mai multor obiecte să se miște liber în scenă. De asemenea, doresc să introduc elemente precum ploaia sau vântul, care vor influența mișcarea frunzelor copacilor, contribuind la realismul scenei.

Un alt aspect pe care aş dori să-l abordez în dezvoltările ulterioare este diversificarea obiectelor din scenă, creând o mulțime mai variată și captivantă. În plus, voi explora posibilitatea de a extinde planul în anumite zone pentru a permite plasarea unor obiecte suplimentare și pentru a îmbogăți experiența vizuală.

5 Referințe

- <https://www.humus.name/index.php> - sursă SkyBox
- <https://www.cgtrader.com>, <https://free3d.com> – surse obiecte 3D
- https://youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-0SM&si=fpG-D7YtNKR5_ITR – tutorial pentru început de Blender