Logo, icon

Description automatically generated

Catedra de Calculatoare și Tehnologia Informației

**DOCUMENTAȚIE**

Prelucrare Grafică

***Proiect: Prezentare Fotorealistă***

***a unei scene de obiecte 3D***

Nume Student: Boloș Andrei-Nicolae

Grupa: 30236

***CUPRINS***

[**Titlu: DOCUMENTAȚIE** 1](#_Toc124944436)

[***CUPRINS*** 2](#_Toc124944437)

[***1.*** ***Rezumat*** 3](#_Toc124944438)

[***2.*** ***Prezentarea scenei*** 3](#_Toc124944439)

[***3.*** ***Detalii de implementare*** 4](#_Toc124944440)

[***4.*** ***Interfața grafică, instrucțiuni de folosire*** 5](#_Toc124944441)

[***5.*** ***Concluzii și dezvoltare ulterioară*** 6](#_Toc124944442)

[***6.*** ***Bibliografie*** 6](#_Toc124944443)

1. ***Rezumat***

Obiectivul principal al aplicației reprezintă prezentarea unei scene formată din obiecte 3D.

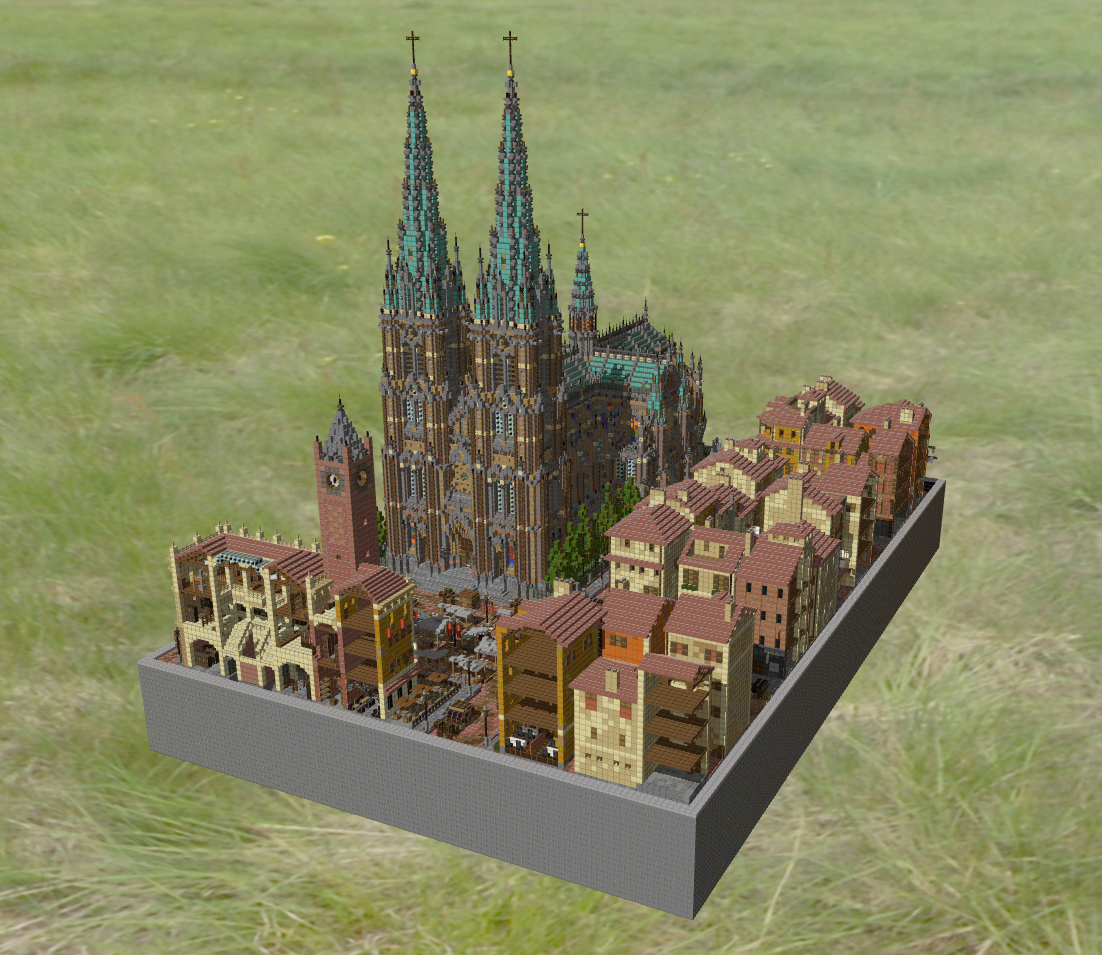
1. ***Prezentarea scenei***

Scena este formată dintr-un centru al unui oraș medieval având ca textură fețele unor block-uri din jocul Minecraft și caracterul principal: Steve.

Aceste două obiecte au fost descărcate de pe site-urile:

* <https://sketchfab.com/3d-models/minecraft-steve-cb228dcc137042cc9a3dc588758cc6e9>
* <https://sketchfab.com/3d-models/plaza-day-time-6a366ecf6c0d48dd8d7ade57a18261c2>

Au fost aduse într-un soft de modelare 3D: Blender. S-au realizat ajustări de scalare și translație pentru a crea scena dorită, iar apoi au fost exportate individual.



1. ***Detalii de implementare***

Realizarea proiectului a început de la fundația oferită de cadrele didactice la laborator. Pe baza acestuia s-au implementat funcționalități prezentate la orele de laborator de pe parcursul semestrului.

Din aceste funcționalități fac parte urmatoarele:

1. Utilizarea structurilor de date folosite în OpenGL *(lab 1, 3 & 4)*;

Exemple:

glm::mat4 model;

glm::vec3 lightDir;

const unsigned int SHADOW\_WIDTH = 2048;

GLint modelLoc;

1. Aplicarea transformărilor (scalare, translație, rotație) unor obiecte*(lab 5)*;

model = glm::translate(model\_steve,glm::vec3(2.0f,1.0f,1.0f));

1. Importarea în scenă a unor obiecte 3D *(lab 6)*;

scena.LoadModel("models/scena/scena.obj");

1. Aplciarea texturilor pe obiecte *(lab 6)*;
2. Adăugarea transformărilor camerei și cea de proiecție *(lab 5)*;

Exemplu de interpretare de utilizare a tastei **W**:

if (pressedKeys[GLFW\_KEY\_W]) {

myCamera.move(gps::MOVE\_FORWARD, cameraSpeed);

//update view matrix

view = myCamera.getViewMatrix();

myBasicShader.useShaderProgram();

glUniformMatrix4fv(viewLoc, 1, GL\_FALSE, glm::value\_ptr(view));

// compute normal matrix for teapot

normalMatrix = glm::mat3(glm::inverseTranspose(view\*model));

}

void Camera::move(MOVE\_DIRECTION direction, float speed) {

if (direction == MOVE\_FORWARD) {

cameraPosition = cameraPosition + speed \* cameraFrontDirection;

}

1. Adăugarea iluminării *(lab 7)*;

vec3 color = min((ambient + (1.0f - shadow) \* diffuse) + (1.0f - shadow) \* specular, 1.0f);

1. Adăugarea umbrelor *(lab 9)*;
2. Adăugarea unui Skybox *(lab 10)*;

std::vector<const GLchar\*> faces;

faces.push\_back("skybox/v2/posx.jpg");…

1. Adăugarea efectului de ceață *(lab 11)*

if(fogEn==1)

fColor =fogColor \* (1-fogFactor) + vec4(color,0.0f) \* fogFactor;.

1. ***Interfața grafică, instrucțiuni de folosire***

În prima ipostază, viziunea scenei este prezentată printr-o vedere de sus a camerei. Utilizatorul se poate plimba prin scenă folosind tastele:

* **W** pentru a merge in față;
* **A** pentru a merge in stânga;
* **S** pentru a merge in spate;
* **D** pentru a merge in jos;
* **Left Shift** pentru a merge in jos;
* **Space Bar** pentru a merge in sus;
* **Scroll Wheel** pentru a mări sau a micșora;

Pentru vizualizarea scenei în diferite moduri, sunt disponibile tastele:

* **B** pentru vizualizare în mod *solid*;
* **N** pentru vizualizare în mod *wireframe*;
* **M** pentru vizualizare în mod *poligonal*;

Pentru vizualizarea efectului de ceață: **F**

Aș fi dorit să implementez o animație pentru modelul *Steve* precum animația prezentată la link-ul ***3***.

1. ***Concluzii și dezvoltare ulterioară***

Acest proiect a avut efectul de a mă învăța cum se dezvoltă aplicațiile grafice interactive.

Aș fi dorit să implementez o animație care se efectuează automat la apăsarea unei taste: să ii aplic modelului Steve niște transformări de rotație în jurul unor puncte exterioare. Aceste operații sunt descrise în imaginea de mai jos. Au la bază animația din video-ul de la link-ul ***3*** din *Bibliografie*.



1. ***Bibliografie***
2. <https://moodle.cs.utcluj.ro/course/view.php?id=524>
3. <https://www.glfw.org/docs/3.3/input_guide.html>
4. <https://www.youtube.com/watch?v=q25fRoACp5I&ab_channel=%D0%9Cr.creeper>
5. <https://www.humus.name/index.php?page=Cubemap&item=Meadow>