A logo with black and red colors

Description automatically generated

# Facultatea de Automatică și Calculatoare

## Prelucrare grafică

Prezentarea fotorealistă a unei scene de obiecte 3D

Mihnea Adriana – Maria

Grupa 30236

Cuprins

[1. Prezentarea temei 3](#_Toc156346363)

[2. Scenariul 3](#_Toc156346364)

[2.1 Descrierea scenei și a obiectelor 3](#_Toc156346365)

[2.2 Functionalitati 5](#_Toc156346366)

[3. Detalii de implementare 5](#_Toc156346367)

[3.1 Functii si algoritmi 5](#_Toc156346368)

[3.1.1 Camera 5](#_Toc156346369)

[4. Modelul grafic 6](#_Toc156346370)

[5. Ierarhia de clase 7](#_Toc156346371)

[6. Interfata grafica + manual de utilizare 7](#_Toc156346372)

[7. Concluzii si dezvoltari ulterioare 8](#_Toc156346373)

[8. Referinte 8](#_Toc156346374)

# Prezentarea temei

Obiectivul temei a fost de a crea o scenă foto-realistă a obiectelor 3D utilizând Biblioteca OpenGL și de a-i permite utilizatorului să inspecteze scena și să interacționeze cu obiectele din ea, utilizând tastatura sau mouse-ul.

Principalele obiective vizate de acest proiect au fost:

* Vizualizare scena: scalare, traducere, rotaţie, mişcarea camerei, animaţii
* Vizualizarea in modurile solid, wireframe, polygonal, smooth
* Specificarea a cel puțin două surse de lumină diferite
* Maparea texturilor
* Umbre
* Animarea obiectelor
* Fotorealism, modelare detaliată, complexitatea scenei, dezvoltarea algoritmului și implementarea

Tema proiectului este o scenă din natură, care combină aceste obiective, în scopul de a construi un mediu foto-realist, și oferă posibilitatea utilizatorului a o vizualiza, a se misca in scena, precum si a interactiona cu luminile și efectele implementate.

# Scenariul

## Descrierea scenei și a obiectelor

Scena este alcatuită dintr-o curte aflată intr-un cadru natural. Aplicatia Blender a fost folosita pentru a importa si modela obiectele din scena.

A house with trees in front of it

Description automatically generated

Scena este compusa din diferite obiecte, aranjate folosind editorul Blender. Obiectele sunt texturate pentru a crea o vedere mai realista. Fiecare obiect vine cu propriul sau fisier de material (.mtl), care descrie materialele aplicate fiecarei parti a acestuia, culoarea si proprietatile de lumina, incluzand de asemenea referinta spre imaginea de textura. Principalele obiecte care compun scena sunt:

* Iarba: un simplu plan sculptat cu forme de relief
* Cerul: skydome făcut cu ajutorul editorului Blender si texturat corespunzator
* Casa: alcatuita din 2 materiale diferite, obiect separat pe materiale
* Dinozaur: alcatuit din 3 materiale diferite, obiect separat pe materiale
* Copaci: separati dupa trunchi si coroana

Toate aceste obiecte pot fi descarcate gratuit de la unul din link-urile furnizate in referinte.

Pentru a completa scena si a o face mai apropiata de realitate, au fost adaugate unele surse de lumina si efectul de ceata. Sursele de lumina pot fi schimbate aplicand scenei diferite fitre.

* Lumina directionala: lumina originala, de zi. Daca este dezactivata, scena devine intunecata
* Lumina punctiforma: o lumina de culoare albastra plasata la un anumit punct in scenaA house with a tree in front of it

  Description automatically generated
* Efectul de ceata

A house with a tree in front of it

Description automatically generated

## Functionalitati

Utilizatorul are posibilitatea sa:

* exploreze si sa vada scena cu ajutorul camerei, ca si cum ar fi in interiorul ei
* activeze/dezactiveze de la tastatura cele 2 surse de lumina si efectul de ceata
* vizualizeze scena in modurile solid, wireframe, polygonal si smooth

A wireframe of a house

Description automatically generated

# Detalii de implementare

Incepand de la template-ul pus la dispozitie in laborator, care include incarcarea unui obiect 3d simplu si a unei lumini directionale de baza, mai multe functionalitati au fost adaugate pentru a-l extinde.

## Functii si algoritmi

### Camera

Pentru a controla miscarile camerei, metodele din clasa Camera au fost implementate astfel incat aceasta sa se miste in toate cele 6 directii: sus, jos, stanga, dreapta, inainte, inapoi.

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

float velocity = speed;

glm::vec3 cameraFront = glm::normalize(cameraTarget - cameraPosition);

glm::vec3 cameraRight = glm::normalize(glm::cross(cameraFront, cameraUpDirection));

switch (direction) {

case MOVE\_RIGHT:

this->cameraPosition.x += speed;

this->cameraTarget.x += speed;

break;

case MOVE\_LEFT:

this->cameraPosition.x -= speed;

this->cameraTarget.x -= speed;

break;

case MOVE\_FORWARD:

this->cameraPosition.z -= speed;

this->cameraTarget.z -= speed;

break;

case MOVE\_BACKWARD:

this->cameraPosition.z += speed;

break;

case MOVE\_UP:

cameraPosition += cameraUpDirection \* speed;

break;

case MOVE\_DOWN:

cameraPosition -= cameraUpDirection \* speed;

break;

}

}

De asemenea, camera poate fi rotita cu un anumit unghi sau in jurul axei Z.

void Camera::rotate(float pitch, float yaw) {

glm::vec3 front;

front.x = cos(glm::radians(yaw)) \* cos(glm::radians(pitch));

front.y = sin(glm::radians(pitch));

front.z = sin(glm::radians(yaw)) \* cos(glm::radians(pitch));

cameraFrontDirection = glm::normalize(front);

cameraRightDirection = glm::normalize(glm::cross(cameraFrontDirection, glm::vec3(0.0f, 1.0f, 0.0f)));

cameraUpDirection = glm::normalize(glm::cross(cameraRightDirection, cameraFrontDirection));

}

# Modelul grafic

Principalele structuri de date utilizate in proiect erau cele definite in librariile OpenGL, precum si cele din locatiile variabilelor uniforme.

* Vectori: glm::vec3, glm::vec4
* Matrici: glm::mat3, glm::mat4
* Valori primitive: GLuint, GLfloat

# Ierarhia de clase

Proiectul este structurat in urmatoarele clase:

* Main
* Camera
* Shader
* Model3D
* Mesh

# Interfata grafica + manual de utilizare

Atunci cand aplicatia este rulata, incepe o animatie de prezentare de cateva secunde alcatuita dintr-o serie de operatii de miscare si rotatie a camerei, astfel incat utilizatorul sa poata avea o vedere de ansamblu asupra scenei.

1. Mod vizualizare:

* Solid: **Y**
* Wireframe: **T**
* Polygonal: **G**
* Smooth: **H**

1. Controlarea miscarilor camerei:
   * + In fata: **W**
     + In spate: **S**
     + Stanga: **A**
     + Dreapta: **D**
     + Sus: **U**
     + Jos: **J**
     + Rotatie dupa axa Y: **Q, E**
2. Controlul surselor de lumina:
   * Lumina directionala: **K**
   * Lumina punctiforma: **L**
3. Controlul efectului de ceata: **F**

# Concluzii si dezvoltari ulterioare

Utilizarea librariei OpenGL ofera o gama larga de posibilitati de a prelucra obiecte si scene 3d folosind simple transformari si functii matematice, cum ar fi animarea unui obiect sau simularea miscarii unei camere. Maparea texturilor a dat obiectelor o nota realista, iar tehnicile de implementare a luminilor au oferit o buna vedere de ansamblu a modului in care obiectele interactioneaza cu lumina in lumea reala.

Ca si extindere a aplicatiei, as fi implementat generare umbrelor si animatiilor separate specifice obiectelor, precum si functii suplimentare precum detectia coliziunilor, ploaie, vant, fulgere.

# Referinte

1. Blender editor: https://www.blender.org/
2. Free 3D obj models:

https://www.cgtrader.com/

https://free3d.com/

1. Texturi pentru skybox:

https://www.humus.name/index.php?page=Textures

1. Referinte pentru implementarea diferitor tipuri de lumini:

https://learnopengl.com/Lighting/Light-casters

1. Suportul de laborator