A picture containing logo

Description automatically generated**FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE**

**SPECIALIZAREA CALCULATOARE SI TEHNOLOGIA**

**INFORMATIEI**

**Documentatie Proiect**

**Prelucrare Grafica**

**Student: Iamnitchi Bogdan**

**Data: 23.12.2023**

**Grupa: 30235**

CUPRINS

[1. PREZENTAREA TEMEI 3](#_Toc92917554)

[2. SCENARIUL 3](#_Toc92917555)

[Descrierea scenei si a obiectelor 3](#_Toc92917556)

[Functionalitati 4](#_Toc92917557)

[3. DETALII DE IMPLEMENTARE 5](#_Toc92917558)

[Functii si algoritmi 5](#_Toc92917559)

[Modelul grafic 6](#_Toc92917560)

[Structuri de date 6](#_Toc92917561)

[Ierarhii de clase 7](#_Toc92917562)

[4. PREZENTAREA INTERFETEI GRAFICE 7](#_Toc92917563)

[5. MANUAL DE UTILIZARE 8](#_Toc92917564)

[6. CONCLUZII 9](#_Toc92917565)

[7. REFERINTE 9](#_Toc92917566)

# PREZENTAREA TEMEI

**Tema Proiectului**: Dezvoltarea unei Scene OpenGL Interactivă

Proiectul constă în dezvoltarea unei scene interactive în OpenGL, în care utilizatorul poate explora un mediu 3D detaliat, dar nu numai ne invata cum proceseaza calculatorul o scena de obiecte, in acelasi timp ne ajuta sa invatam sa folosim diferite biblioteci si librarii 3D, cum ar fi OpenGl, GLFW, GLM. Scena include diverse obiecte, precum castele, clădiri, personaje, și efecte speciale, iar utilizatorul acestei aplicatii are posibilitatea sa se plimbe prin scena cu ajutorul tastaturii dar si a mouse-ului, exact ca si intr-un joc pe calculator.

Obiective acestui proiect au fost:

1. Crearea unei scene 3D detaliate și interactivitate cu obiectele acesteia.
2. Implementarea unui sistem de camere zburătoare pentru explorarea scenei.
3. Adăugarea de modele animate și sunete pentru a crea o experiență captivantă.
4. Integrarea de efecte vizuale, precum umbre, ceață și lumini dinamice.
5. Optimizarea performanței pentru a menține o rată bună de cadre pe secundă.

Tehnologii Utilizate:

* OpenGL pentru randare 3D.
* C++ pentru implementarea logicii de program.
* Modelul de iluminare Blinn-Phong pentru obținerea efectelor vizuale realiste.
* Utilizarea de shaderi pentru a controla aspecte specifice ale rendering-ului.

Resurse Adiționale:

* Biblioteci sau framework-uri specifice pentru gestionarea resurselor grafice.
* Sursa melodiilor de fundal și sunetelor utilizate în proiect.

# SCENARIUL

## Descrierea scenei si a obiectelor

În universul fermecat al acestei scene OpenGL, fiecare detaliu este sculptat cu grijă pentru a aduce la viață o poveste captivantă. La baza acestei lumi magice, se întinde un mozaic tridimensional al diverselor elemente care converg pentru a crea o experiență cu adevărat remarcabilă.

Scena reprezinta un scenariu medieval dar cu mici interventi din lumea SF (Științifico-fantastic), precum dragonii care scuipa foc. La baza scenei se afla un teren de munte facut de mine personal in blender folosind Sculpt Mode, pe care am mapat diferite texturi de iarba pentru a fi mai realistic. Dupa am adaugat caselul principal care constituie din diverse alte obiecte care m-am asigurat ca au aplicate un singur material pentru a fi mapate in aplicatia mea. Dupa am adaugat apa care este un plan pe care am aplicat o textura, iar dupa au urmat diferse alte obiecte. Am generat multe alte castele in care a trebuit sa mapez individual fiecare textura. Am generat un drum (folosind modifier-ul array si curve din blender) care conecteaza majoritatea cladirilor, precum si un mic satuc izolat langa apa si un far pe o insula indepartata. Scena mai contine diferite obiecte interactive precum elicea de la moara care se invarte mereu, vulturi din jurul castelului se invart si scot sunete, dragonii pot scuipa mingi de foc si folosesc de asemenea efecte sonore, si mai exista o barca de pirati care se plimba pe apa intr-un loop continuu, toate aceste animati pot fi oprite si pornite folosind tastatura. Putem controla si sursele de lumina precum si activarea sau dezactivarea efectului de ceata. In cele din urma mai exista si o animatie de prezentare in care camera se plimba singura prin scena pentru a prezenta fiecare obiect aflat in scena.

Desigur scena poate fi vizualizata in diferite moduri cum ar fi smooth sau fill, wireframe si point.

## Functionalitati

Pentru a putea vizualiza scena, aceasta este luminata de o lumina directionala, dar si una punctifroma stituata pe farul de pe insula. Persoanele care doresc sa testeze acest joc, se pot deplasa prin scena cu ajutorul tastelor W, A, S, D, dar si cu ajutorul mouse-ului. Scena fiind foarte mare am implementat si functia de zoom care poate fi controlata cu ajutorul rotitei de scroll de pe mouse. Se poate creste si scade viteza camerei folosind butaonele +,- (keypad)

Exista si o animatie de prezentare in care camera se misca singura dupa diferite curbe descrise matematic in cod, si cuprinde vizualizarea fiecarui obiect din scena.

Utilizatorul are un vast control asupra randari scenei, poate activa si dezactiva atat lumina punctiforma cat si cea directionala utilizand tastele 8 si 9. Poate de asemenea activa sau dezactiva umbrele folosind tasta 0. Apasand anumite taste, utilizatorul poate creea anumite chiar si situatii meteo precum ar fi ceata pe tasta 7, in acelasi timp putand vizualiza in diferite moduri scena apasand pe tastele 1, 2, 3.

Pentru umbre am implementat tehnica prezentata in laborator, shadow mapping deoarece este cea mai usor de implementat in OpenGL. Pentru a roti lumina si a vedea efectul de umbram trebuie sa apasam pe butoanele J sau L, deoarece numai in acest mod lumina se va roti si va da o senzatie de umbra. Pentru a vizualiza harta de adancime trebuie doar sa apasam pe tasta TAB.

Am implementat si diferite animatii care fac ca scena sa aiba mai multa viata si sa para cat mai realista:

* Elicea de la moara se invarte la intrarea in scena, se poate opri si porni de pe tasta M
* Barca cu pirati se plimba pe apa intr-un sens si in altul, insotit de o melodie, se poate opri sau porni de pe tasta 8 din sectiunea tastaturi keypad.
* Vulturi zboara in jurul castelului si folosesc diferite efecte sonore, acestia pot fi opriti sau porniti de pe tasta 5 (keypad)
* Dragonii ce pazesc castelu pot scuipa mingi de foc acestia sunt doi la numar si pot fi fiecare controlati de pe tastele 4 si 6 (keypad)
* Caracterul principal care este un luptator medieval este randat initial simplu fara arme, dar poate fi inarmat folosind tasta 2 (keypad). Poate trece si in pozitie de atac folosind tastele 1 respectiv 3 (keypad)
* Putem roti caracterul cu tastele Q si E dar si lumina de pe tastele J si L pt a vizualiza mai bine umbrele

# DETALII DE IMPLEMENTARE

## Functii si algoritmi

Librariile pentru OpenGL detin o multime de functii ajutatoare asa ca o sa incepem descrierea prima, in ordinea aparitiilor in proiect. Astfel prima functie este cea care initializeaza fereastra pe care se va proiecta harta, care are ca si parametrii dimensiunea de vizualizare a ferestrei, dar si un nume: myWindow.Create(glWindowWidth, glWindowHeight, "OpenGL Project Core").

Deoarece pe ecran desenam punct cu punct, si fiecare din ele se proceseaza in shadere, putem spune ca una din cele mai importante functii ale acestui proiect este renderScene(), deoarece aici desenam obiectele in functie de parametrul TAB de intrare, daca el nu a fost apasat desenam scena totala, iar daca nu desenam doar mapa de adancime sa vedem cum se deseneaza umbrele pe harta.

Funcția initModels() joacă un rol esențial în pregătirea scenei, gestionând încărcarea și plasarea modelelor. Ecranul Quad servește scopuri de diagnosticare, prezentând harta adâncimii. Castele, clădiri, harta și satul adaugă stabilitate și detaliu peisajului, în timp ce personajul warrior și obiectele sale mobile, precum și elementele luminoase și speciale, aduc viață și mișcare în scenă.

Scenele statice și mobile sunt pătrunse de iluminare atât punctiformă, cu obiecte precum lumina castelului și a insulei, cât și direcțională, modelată prin lumina soarelui. Shaderul Blinn-Phong este folosit pentru a obține reflexii și iluminări realiste, contribuind la aspectul natural și coerent al scenei.

O alta functie importanta este initUniforms() deoarece aici setam adresa locatiilor variabilelor aflate in shadere, dar si le initializam cu o valoare anume, spre exemplu cu detalii despre lumina, cu perspectiva camerei, matricea de normale. Fara ea programul nu ar putea functiona correct. Tot aici vreu sa specific si functia initShaders() unde incarcam continutul shaderelor. Fiecare obiect se va desena de obicei prin intermediul unu shader.

Pentru a putea implementa miscarea camerei prin intermediul tastelor si a mouse-ului, am fost nevoit sa implementez mouseCallback() dar si processMovement(). Am facut acest lucru pentru a se putea vizualiza scena mult mai usor. Dar si scrollCallback() pentru a putea implementa dunctia de zoom in scena.

Pentru fundal am implementat si un sky box ca sa para ca lumea este infinita chiar daca ea nu se termina niciodata. Am incarcat acest sky box cu ajutorul renderSkyBox(), dar si de implementarea SkyBox.h oferita la laborator. Skybox ul este fur si simplu un cub fromat din mai multe poze. Cel folosit din proiect este luat din unul din site urile specificate in laborator.

Pentru a genera sunete in proiectul meu, am apealat al LearnOpenGl unde am aflat ca pot folosi libraria IrrKlang care vine gata precompilata doar trebuie sa o folosim. Odata importata libraria trebuie doar sa cream un Enigine de sunet folosind

**engine = irrklang::createIrrKlangDevice();**

iar dupa doar apelam:

**engine->play2D("sounds/dragon\_fire.wav", GL\_FALSE, GL\_FALSE, GL\_TRUE);**

trebuie sa avem grija ca la final sa eliberam memoria alocata pt engine deci:

**engine->drop();**

Pentru a genera ceata am folosit sursele din laborator unde am folsoit functia din shadere clamp() care se asigura ca o valoare data este intr un range specificat. Si functia mix() care interpoleaza linear o valoare intre alte doua specificate. Pentru diferitele surse de lumina am folsit cate un struct ca sa tin lucrurile organizate.

Calendar

Description automatically generated

## Modelul grafic

Harta a fost generata integral de mine in blender utilizandu-ma de subdivide pe un plan dupa folosind sculpt mode am generat deformarile pe care mi-le am dorit. Am zis ca ar fi dragut sa am si o apa dar ca sa nu consum multe resurse am recurs la un plan care intretaie pamantul deja generat si umple toate golurile la acelasi nivel cum ar face apa in mod real. Am aplicat o textura pentru apa ca sa fie cat mai realist si zic eu a iesit minunat. A urmat o serie de obiecte care au fost descarcate de pe diferite site uri. Dar nici una din ele nu au venit cu texturile bine mapate pe obiect asa ca pe fiecare obiect din scena am folosit UV editor si am mapat textura asa cum am stiut eu mai bine. Am avut grija ca obiectele pe car ele iau sa fie low polly ca sa optimizez cat mai tare proiectul, abatere de la acest lucru face castelul principal care are aproape 1mil de triunghiuri dar mi-a placut mult si am zis ca merita. Ca sa fie si mai realist am zis sa adaug niste drumuri ce leaga clastelul de alte cladiri sau alte castele acest lucru l-am facut manual in blender folosindu-ma de modifier-ul array si curve. Desigur si toata aranjarea in scena a fost facuta in blender pentru a imi usura munca dar numeroase obiecte au fost importate si single pt a fi manipulate de mine in program (vulturi, elice moara, warrior, armele, curburile pt sursele de lumina, dragonii, bilele de foc).

## Structuri de date

Printre singurele structuri de date utilizate este cea din clasa de camera unde avem nevoie de mai multi vectori, cum ar fi pozitia camerei, directia de vizualizare, dar si coordonata de pozitie a camerei. Alte structuri de date importante sunt Glint in care sunt stocate adresele variabilelor uniform din shadere, dar si vectori de diferite dimensiuni si marimi. Nu in ultimul rand am folosit data de tipul Model sau Shaders dar si tipuri normal din C++ cum ar fi bool, int, char.

Am adus unele mici modificari la obicetul camerei deoarece am implementat zoom si am mutat unele lucruri specifice camerei din main in camera.cpp, precum am creat diferite gettere si settere pentru accesul cu usurinta la vectori camerei.

Am folosit 3 structuri in basic.frag care reprezinta materialul care poate avea mai multe componente precum textura si shineness, lumina directionala cu directie componenta ambineta difuza si speculara. Precum si lumina punctiforma care are pozitie, componentele ambientala difuza si speculara dar si coeficienti de atenuare constant liniar si cuadratic.

## Ierarhii de clase

In afara de main fiecare clasa are cate un header. Urmeaza acum sa dau exemplu de clase si fiecare ce face:

main.cpp:

Punctul de intrare în program, unde este creată instanța Window și se inițializează restul componentelor.

* Camera.cpp și Camera.hpp:

Clasa Camera gestionează aspectele legate de perspectiva și poziționarea camerei în spațiu. Include funcționalități pentru mișcare și controlul camerei.

* Mesh.cpp și Mesh.hpp:

Clasa Mesh reprezintă o rețea de triunghiuri care definește geometria unui obiect 3D. În această clasă se pot stoca și manipula vertecșii, normalele și coordonatele texturilor.

* Model3D.cpp și Model3D.hpp:

Clasa Model3D gestionează un obiect 3D format din una sau mai multe plase (Mesh). Aici se pot adăuga funcționalități pentru animație, manipulare și afișare a modelelor.

* Shader.cpp și Shader.hpp:

Clasa Shader oferă un mecanism pentru gestionarea shader-elor OpenGL, inclusiv încărcarea, compilarea și activarea lor în cadrul programului OpenGL.

* SkyBox.cpp și SkyBox.hpp:

Clasa SkyBox se ocupă de afișarea și gestionarea unei cutii cerești, adesea folosită pentru a crea un fundal înscene 3D. Include funcționalități pentru încărcarea și randarea texturilor cubice.

* stb\_image.cpp și stb\_image.hpp:

Biblioteca stb\_image furnizează funcții pentru încărcarea de imagini. Este adesea folosită pentru încărcarea texturilor în cadrul proiectelor OpenGL.

* tiny\_obj\_loader.cpp și tiny\_obj\_loader.hpp:

Biblioteca tiny\_obj\_loader este utilizată pentru încărcarea modelelor în formatul .obj. Oferă funcționalități pentru citirea și interpretarea datelor din fișierele .obj.

* Window.cpp și Window.hpp:

Clasa Window reprezintă fereastra aplicației și oferă funcționalități pentru gestionarea evenimentelor de input, inițializarea contextului OpenGL și afișarea graficelor.

# PREZENTAREA INTERFETEI GRAFICE

**A picture containing graphical user interface

Description automatically generatedCalendar

Description automatically generatedA screenshot of a video game

Description automatically generated with medium confidenceModuri de vizualizare:**

**Calendar

Description automatically generatedA picture containing calendar

Description automatically generatedCu si fara fragment discarding**:

**Calendar

Description automatically generated with medium confidenceA picture containing text, electronics, display

Description automatically generatedCu si fara ceata**:

# MANUAL DE UTILIZARE

Utilizatorul se poate bucura de urmatoarle functionalitati implementate si legate de tastele:

Camera:

* W - deplasare camera in fata
* S - deplasare camera in spate
* A - deplasare camera in stanga
* D - deplasare camera in dreapta
* UP – deplasare camera in sus
* DOWN – deplasare camera in jos
* SPACE – porneste / opreste animatia de perezentare a scenei
* + (keypad) – creste viteza camerei cu 1.0f
* - (keypad) – scade viteza camerei cu 1.0f
* Miscarea mouse-ului genereaza rotatia camerei yaw si pitch (restrictionat la 89.0f grade)
* Miscarea rotitei de scroll face zoom in / zoom out

Lumini si Umbre:

* J - deplasare lumina in cerc (in partea dreapta)
* L - deplasare lumina in cerc (in partea stanga)
* TAB – afisarea hartii de adancime
* 8 – porneste / opreste lumina globala (directionala)
* 9 - porneste / opreste lumina locala (punctiforma)
* 0 – porneste / opreste generarea umbrelor asupra pesonajului

Vizualizare scena

* 1 – vizualizarea scenei in modul solid
* 2 – vizualizarea scenei in modul polygonal
* 3 – vizualizarea scenei in modul point
* 7 – porneste / opreste ceata

Animatii:

* M – porneste / opreste rotatia elicei de pe moara
* Q – roteste personaju principal la dreapta
* E – roteste personaju principal la stanga
* 8 (keypad) - porneste / opreste animarea miscarii barci cu pirati impreuna cu muzica
* 5 (keypad) - porneste / opreste animarea vulturilor ce zboara in jurul castelului cu efecte sonore
* 4 (keypad) – dragonul din stanga scuipa o minge de foc
* 6 (keypad) – dragonul din drepata scuipa o minge de foc
* 2 (keypad) – personaju principal este inarmat / dezarmat cu scut si sulita
* 1 (keypad) – personajul trece in pozitie de atac rotind sulita si apropind scutu
* 3 (keypad) – personajul trece in pozitie normala de prezentare

# CONCLUZII SI DEZVOLTARI ULTERIOARE

Acest proiect a servit nu doar ca o platformă pentru a aplica cunoștințele în programarea OpenGL, ci și pentru a înțelege profunzimile interacțiunii dintre modelare, animație și programare. Integrarea obiectelor create în Blender a adăugat o dimensiune suplimentară de complexitate, necesitând abordări creative și soluții tehnice pentru a le încorpora armonios în cadrul proiectului OpenGL.

Procesul de a aduce la viață fiecare detaliu al scenei tale a necesitat nu doar cunoștințe tehnice, ci și abilități de rezolvare a problemelor și gândire creativă. De la gestionarea resurselor până la optimizarea performanței și implementarea animațiilor din cod, această experiență a oferit o perspectivă cuprinzătoare asupra dezvoltării unui proiect 3D complex.

Cu fiecare dificultate întâlnită și depășită, ai acumulat nu doar competențe tehnice, ci și înțelegerea necesară pentru a aborda provocările specifice în domeniul graficii 3D. Acest proiect nu doar reflectă realizările tale în dezvoltarea software-ului, ci și creșterea ta personală ca programator și creator de experiențe interactive.

Ca si dezvoltari ulterioare s-ar putea anima personajul ca sa se poata misca folsoind skeleton animation, dar si implementarea unei camere Third Person Camera care mi-am dorit-o dar nu a fost neaparat scopul acetui proiect. Aceasta scena pune chiar bazele unui joc 3D si poate chiar fi asa dezvoltat, sa zicem ca pe acel pod mare vin alti adversari pe care tu trebuie sa-i omori ca sa poti ajunge la castel sa salvezi o printesa dar si castelu e pazit de dragoni care o sa incerce sa te omoare si o sa i poti omora folosind tunurile de pe turnuri.

# REFERINTE

1. <https://learnopengl.com/> [pentru a invata bazele pt OpenGL, decembrie 2023]
2. <https://free3d.com/> [pentru diferite obiecte, decembrie 2023]
3. <https://www.turbosquid.com/> [pentru diferite obiecte, decembrie 2023]
4. <https://www.cgtrader.com/> [pentru diferite obiecte, decembrie 2023]
5. [Tutoriale Blender –](https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM) Constantin Nandra [pentru bazele blender, decembrie 2023]
6. [Blender Tutorial](https://docs.google.com/document/d/1njtWPMmOQNIaD_z9ve8iPRUqQTWdIV_PO-NvPD0nOuM/edit#heading=h.7ck7keotfet) - Student Mix [baze blender dar si tips and tricks, decembrie 2023]