A picture containing logo

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

**CATEDRA 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 acestui proiect este de a ne ajuta sa ne familiarizam mult mai bine cu fotorealistica in jocuri, 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. Utilizatorul acestei aplicatii trebuie sa aibe posibilitatea sa se plimbe prin scena cu ajutorul tastaturii dar si a mouse-ului, exact ca si intr-un joc pe calculator. Pentru scena proiectului, am ales sa ma folosesc de o scena gata creeata, eu doar sa aranjez obiectele asa cum vreau. A durat foarte mult texturarea lor dar spun ca a meritat.

# SCENARIUL

## Descrierea scenei si a obiectelor

Scena de obiecte reprezinta un colt de natura rupta dintr-un basm, mai exact o padure cu un cer senin, ca asa am vrut eu sa fie cer senin, ca de altfel se putea implementa orice alt fel de cer. Exista mai multe carari care duc in diverse locatii de pe harta. Exista un o zona in care locuiest personajul principal al acestui basm, in zona de corturi, o locuinta ferita de soare la umbra unor copaci. Creearea scenei a durat destul de mult deoarece a fost nevoie sa texturez fiecare obiect, si il mut in zona care mi-a placut mie, si sa il rotesc ca sa arate bine. Pan sa aflu de fragment discarding, ramaneam de fiecare data dezamagit cand importam scena in OpenGL, deoarece in Blender harta arata atat de bine dar in VisualStudio nu. Dupa ce am implementat fragment discarding fucrurile au inceput sa stea mult mai bine si sa arate ca si o scena reala. Miscarea umbrelor impreuna cu lumina directionala se realizeaza de pe tastele Q si E, iar ceata se poate activa de pe tasta V si dezactiva de pe tasta B.

A picture containing calendar

Description automatically generated

## Functionalitati

Pentru a putea vizualiza scena, aceasta este luminata de o lumina punctiforma, care se deplaseaza prin scena pentru a demonstra implementarea umbrelor. 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. Apasand anumite taste, utilizatorul poate creea anumite situatii meteo, in acelasi timp putand vizualiza in diferite moduri scena apasand pe tastele Z, X, C.

Pentru a para cat mai bine texturata scena si cat mai reala am implementat fragment discarding, pentru ca frunzele copacilor sa nu aiba un fundal prost texturat, asa parand totul mult mai realis. Tot pentru realism am implementat si ceata pe care o porenesti de pe tasta V si o opresti de pe tasta B.

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 E sau Q, 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 M.

# 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 la mine 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 M 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.

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, c matricea de normale. Fara ea programul nua r putea functiona deloc correct. Tot aici vreu sa specific si de functiile initModels() cu ajutorul careia importam in OpenGL un obiect 3D, dar si 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.

Pentru a nu parea atat de urat fundalul 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(), acesta fiind cel prezenta in laborator, cu cerul senin, soarle pe cer dar si cu munti in fundal.

Calendar

Description automatically generated

## Modelul grafic

Harta si obiectele nu sunt creatie proprie ci sunt decarcate de pe diferite siteuri de pe net. De mine sunt doar aranjate si si texturate in Blander. Am facut acest lucru ca sa imi usurez munca, deoarece sunt peste 100 de obiecte si daca importam fiecare obiect pe rand trebuia sa fac o multime de operatii pe el, asa mi-am facut viata mult mai usoara, iesind zic eu cu mult mai realist si mai frumos asa.

## 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. Variabilele normale au fost utilizate pentru a putea implementa diverse functionalitati.

## Ierarhii de clase

In afara de main fiecare clasa are cate un header. Urmeaza acum sa dau exemplu de clase si fiecare ce face. Clasa de camera are implementat in ea camera, clasa de window are imlementate toate functionalitatile pentru creearea si stergerea unei ferestre de desenare, clasa de skyBox incarca, deseneaza si efectueaza diversee operatii pe sky box, clasa de mesh deseneaza obiecte, clasa de shaders fefectueaza toate operatiile necesare implementarii tuturor functionalitatilor unui shader nu in ultimul rand std\_image si tiny\_obj\_loader textureaza respectiv incarca un ubiect in OpenGl.

# 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 generatedCalendar

Description automatically generatedCu si fara umbre:

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 taste:

* V - activarea / B - dezactivarea cetii.
* W - deplasare camera in fata
* S - deplasare camera in spate
* A - deplasare camera in stanga
* D - deplasare camera in dreapta
* E - deplasare lumina in cerc (in partea dreapta)
* Q - deplasare lumina in cerc (in partea stanga)
* M – afisarea hartii de adancime
* Z – vizualizarea scenei in modul solid
* X – vizualizarea scenei in modul smooth
* C – vizualizarea scenei in modul polygonal
* F – porneste / G stinge a doua lumina

# CONCLUZII

Acest proiect vine in ajutorul studentiilor care vor sa invete modelare pri OpenGl, carora le plac jocurile pe calculator si senzatiile tari. Am mai invatat cum se textureaza un obiect, cum se modeleaza un obiect 3D dar si cum se aplica unele transformari pe obiecte. Pentru mine a fost o experienta placuta in a crea prima mea aplicatie implementata ca first person. As vrea sa invat mai multe despre aceste tehnici, unele din ele deosebit de grele pentru persoanele neinitiate in acest domeniu.

# REFERINTE

<https://learnopengl.com/> [pentru a invata bazele pt OpenGL, ianarie 2022]

<https://free3d.com/> [pentru diferite obiecte, ianarie 2022]

<https://wow.tools/export/> [pentru a descarca diferite harti, ianuarie 2022]