A black background with red lines

Description automatically generated

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**SPECIALIZAREA CALCULATOARE**

**PROIECT PG**

Disciplina : Prelucrare Grafică

Student : Pojar Andrei-Gabriel

An : III

Grupa : 30233

Profesor coordonator : Stanca Ostafie

**CUPRINS**

1. Prezentarea temei
2. Scenariul

* Descrierea scenei și a obiectelor
* Funcționalități

1. Detalii de implementare

* Funcții și algoritmi
* Soluții posibile
* Motivarea abordării alese
* Modelul grafic
* Structuri de date
* Ierarhia de clase

1. Prezentarea interfeței grafice utilizator/ manual de utilizare
2. Concluzii și dezvoltări ulterioare
3. Referințe
4. **Prezentarea temei**

Proiectul propus este o aplicație în OpenGL în care se integrează o scenă creată în Blender de mine. Scopul acestei aplicații este să ofere o experiență interactivă utilizatorului într-un mediu tridimensional, să ne ajute să înțelegem cum calculatorul procesează o scenă de obiecte și să utilizăm diverse biblioteci și librării 3D cum ar fi OpenGL, GLFW sau GLM.

Scena mea prezintă o fermă în mediul rural în care apar multe obiecte specifice acestei tematici. Utilizatorul aplicației poate să vadă această scena, să se plimbe prin ea utilizând tastatura sau mouse-ul și poate observa și controla diverse animații implementate.

1. **Scenariul**

* Descrierea scenei și a obiectelor

Scena mea reprezintă o fermă din mediul rural în care apar multe obiecte specifice acestei tematici cum ar fi: hambar, tractor, diverse animale(cal, câine, pisică, curcan,etc), siloz și multe alte elemente specifice. Creearea acestei scene a avut loc în Blender 4.0 unde am descărcat obiecte de pe internet, le-am texturat(unele le-am descărcat gata texturate) și le-am poziționat după bunul plac. Cu ajutorul modului Sculpt Mode din Blender am reușit să fac un mic lac și diferite forme de relief, ca să pară cât mai realistică scena. Fiecare obiect l-am texturat, l-am poziționat unde am vrut eu, după care l-am scalat, rotit și l-am făcut să arate exact cum am vrut. După ce am realizat scena, am început să exportez individual mai întâi obiectele cu care am vrut să fac animații și apoi să exportez întreaga scenă. După ce am facut asta am creat un folder models în care am pus scenaaaaaaaa.obj, scenaaaaaaaa.mtl și toate texturile folosite pentru obiectele din scenă. Apoi m-am asigurat că in .mtl sunt prezente numele texturilor și nu path-ul către acestea. După aceea am încărcat scena în Visual Studio și am început partea de codat.

* Funcționalități

Utilizatorul poate să navigheze prin scenă utilizând atât tastatura cât și mouse-ul. De pe tastele W,A,S,D se realizează mișcarea planului scenei la stânga, dreapta cât și mărirea respectiv micșorarea planului pe care se află scena. Cu ajutorul tastelor de UP, DOWN, LEFT, RIGHT pot muta efectiv poziția camerei care e îndreptată către scena mea. Apoi cu ajutorul tastelor 5,6,7 pot vizualiza scena în modurile SOLID, WIREFRAME și POINTS. De asemenea de pe tastele Q și E pot roti planul pe care se află scena. Apoi cu tasta Y fac cerul să fie pe modul dark iar cu R revin la cerul pe timp de zi. Pentru a roti lumina care bate pe scenă folosim tastele J și L, iar în acest mod putem vizualiza diferite ipostaze ale scenei în funcție de poziția luminii. Pentru partea de realism am implementat ceața care se pornește de pe tasta Z și o oprești de pe tasta X. De asemenea dacă pornești modul de ceață poți să setezi cât de densă să fie cu ajutorul tastei C care o face tot mai densă, iar cu tasta V o faci mai puțin densă. Cu ajutorul tastei M putem vizualiza harta de adâncime însă aceasta mi se afișează în totalitate neagră. De asemenea am realizat două animații: mișcarea la stânga și la dreapta continuă a unui balon cu aer cald, respectiv rotirea ,,dezordonată” a unei drone. Prin mișcare dezordonată m-am referit că în timpul rotirii drona și urcă sau coboară pentru a da realism animației, exact cum se comportă în realitate o dronă care zboară.

1. **Detalii de implementare**

* Funcții și algoritmi

Codul începe prin includerea bibliotecilor și a fișierelor necesare pentru OpenGL, GLFW și GLM precum și alte fișiere sursă și fișiere header definite. Apoi sunt declarate toate variabilele globale pentru diferite matrici de transformare, variabile de shader, modele de obiecte 3D și altele. Există și variabile care controlează modurile de iluminare, modul de prezentare și altele. Voi prezenta pe scurt câteva funcții importante de la mine din proiect: Funcțiile pentru gestionarea evenimentelor de la fereastră(windowResizeCallback,keyboardCallback și mouseCallback), funcții pentru acutalizarea matricelor de vizualizare și normală(updateViewAndNormalMatrix), funcții pentru gestionarea mișcării și interacțiunii cu utilizatorul(processMovement, presentationMode, modes, haze, lightControl), funcții de inițializare în OpenGL( initOpenGLWindow, setWindowCallbacks, initOpenGLState), funcții de inițializare pentru modele și shadere(initModels, initShaders, initUniforms), funcții pentru inițializarea framebuffer-ului pentru shadow mapping( initFBO). Mai sunt funcții pentru calculul și setarea matricelor de tranziție light-space( computeLightSpaceTrMatrix), funcții pentru desenarea obiectelor și gestionarea scenelor( drawObjects, renderScene), funcții pentru gestionarea prezentării (tasteP). Mai sunt funcții pentru curățare(cleanup) și pentru inițializarea skybox-ului(initSkyBox). Mai este partea de main în care apelez funcțiile.

* Soluții posibile

Pentru controlul camerei se utilizează funcții de rotație și translație pentru a actualiza matricea de vizualizare. Apoi prin rotația matricei de lumină controlez lumina din scenă. Prezentarea scenei prin apăsarea unui buton se modifică poziția și orientarea camerei în funcție de unghiul curent al prezentării.

* Motivarea abordării alese

Am ales această abordare deoarece m-am concentrat tare pe experiența pe care utilizatorul o va avea la deschiderea aplicației. Am vrut ca utilizatorul să se bucure de o vizualizare frumoasă, să poată explora prin scenă și să interacționeze cu diverse obiecte/efecte. Am selectat atent platformele și tehnologiile care au contribui mai apoi la eficiența implementării.

* Modelul grafic

Obiectele pe care le-am folosit în scenă le-am descărcat de pe Internet. Le-am descărcat sub format .obj, împreună cu texturile(dacă au avut) și cu .mtl. Le-am adăugat în scenă și cu ajutorul lor am realizat scena. După ce am terminat am exportat scena realizată împreună cu exportul individual al obiectelor care fac obiectul unei animații.

* Structuri de date

Fiecare obiect 3D din scenă este asociat cu informații precum poziția, rotația, scalarea și materialele asociate. În clasa Camera utilizez structuri de date deoarece avem nevoie de mai mulți vectori, cum ar fi poziția camerei, direcția de vizualizare și coordonata de poziție a camerei. De asemenea am utilizat date de tipul Model3D, Shaders, Gluint. Am folosit și tipuri din C/C++ cum ar fi int,bool sau float.

* Ierarhii de clase

Fiecare clasă are câte un fișier header, cu excepție clasei main care e clasa principală a proiectului.

A screenshot of a computer program

Description automatically generated

1. **Prezentarea interfeței grafice utilizator/manual de utilizare**

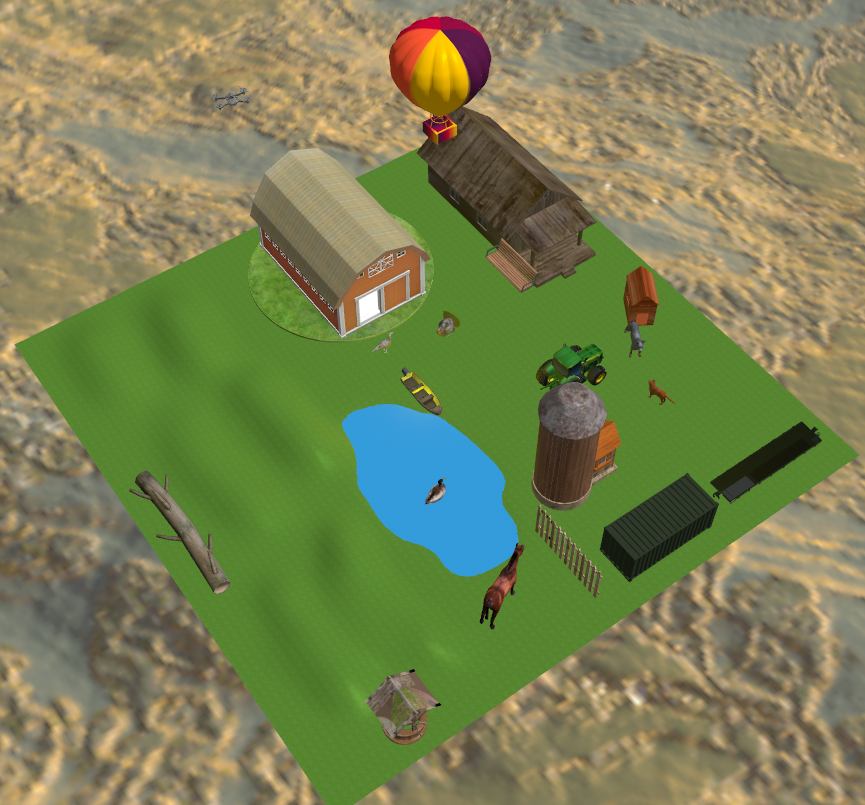
Acesta este scena pe care am realizat-o in Blender și mai apoi am exportat-o in OpenGL.

Prin această scenă realizată vreau să prezint o scenă de la mediul rural de la o fermă. În primul rând am adăugat animale specifice cum ar fi câinele împreună cu coliba lui, calul, pisica, curca și curcanul, mai apoi alte elemente de specifice agriculturii cum ar fi tractorul, hambarul și silozul. Am vrut să fac scena cât mai realistă și de aceea am făcut și un lac pe care stă o rață și lângă o barcă. Bineînțeles am adăugat și o casă care este practic casa fermierului. Am mai adăugat un container în care pot fi ținute diverse unelte, un gard, un trunchi de arbore tăiat și un vagon-platformă. Se poate observa și balonul și drona care fac obiectul celor două animații din scenă.

A screenshot of a video game

Description automatically generated

Moduri de vizualizare: SOLID, WIREFRAME și POINTS



A screenshot of a video game

Description automatically generated

A group of people and a hot air balloon

Description automatically generated

Vizualizarea scenei cu efectul de ceață pornit:

A screenshot of a video game

Description automatically generated

Vizualizarea scenei cu cerul de mod dark:

A screenshot of a video game

Description automatically generated

Manual de utilizare- Utilizatorul se poate bucura de câteva funcționalități implementate și le poate testa cu ajutorul tastaturii și mouse-ului.

Mouse: cu ajutorul mouse-ului poți schimba direcția camerei cu ajutorul căreia poți vizualiza scena

Tastatura: W,A,S,D - se realizează mișcarea planului scenei la stânga, dreapta cât și mărirea respectiv micșorarea planului pe care se află scena

Q,E - pot roti planul pe care se află scena

Z,X,C,V – controlul efectului de ceață(Z-pornirea efectului de ceață, X- oprirea efectului de ceață, C- mărirea densității ceței, V- micșorarea densității ceței)

M – harta de adâncime

Y- modul cerului dark

R – revenire la modul de zi de la modul dark

P – prezentarea scenei

5,6,7 – vizualizarea modurilor SOLID, WIREFRAME și POINTS

UP,DOWN,LEFT,RIGHT - pot muta efectiv poziția camerei care e îndreptată către scena.

1. **Concluzii și dezvoltări ulterioare**

Proiectul dezvoltat a fost un proces de învățare și implementare complex, furnizând o platformă pentru dezvoltarea aplicațiilor interactive 3D. Un aspect esențial al proiectului a fost utilizarea bibliotecilor OpenGL și GLM. Acestea au oferit funcționalități esențiale pentru manipularea grafică. Adăugarea unor elemente precum skybox-ul au contribuit la atmosfera generală a scenei.

Posibilele dezvoltări ulterioare ar fi integrarea materialelor și a texturării avansate care poate duce la un grad mai mare de detaliu și realism. O altă posibilă dezvoltare ar fi adăugarea de sunete și dezvoltarea shaderelor pentru a îmbunătăți calitatea vizuală a scenei adăugând efecte precum reflexii sau efecte de particule.

1. **Referințe**

* Lucrările de laborator și tutorialele pentru Blender
* <https://learnopengl.com/>
* <https://free3d.com/>
* <https://learnopengl.com/Getting-started/Shaders>
* <https://www.lighthouse3d.com/tutorials/glsl-12-tutorial/directional-lights-i/>