A logo with black and red lines

Description automatically generated

Documentație proiect OpenGL Prelucrare Grafică

A cartoon of a house and a dog

Description automatically generated

Student: Marcu Ariana-Mălina

Grupa: 30232/2

CUPRINS

*1. Cuprins  
2. Prezentarea temei  
3. Scenariul  
    3.1. descrierea scenei și a obiectelor  
    3.2. funcționalități  
4. Detalii de implementare  
5. Prezentarea interfeței grafice utilizator / manual de utilizare  
6. Concluzii și dezvoltări ulterioare  
7. Referințe*

1. Prezentarea temei

Proiectul propus vizează crearea unei prezentări fotorealiste a unor scene de obiecte 3D, utilizând biblioteci precum OpenGL, GLFW și GLM. Scopul principal este oferirea utilizatorului posibilității de a controla scena prin intermediul tastaturii și mouse-ului. Proiectul se axează pe diverse aspecte, cum ar fi vizualizarea scenei, specificarea surselor de lumină, moduri de afișare a scenei, maparea texturilor și materialelor, generarea umbrelor, animarea obiectelor și alte aspecte complexe care contribuie la realizarea unui mediu 3D realist.

1. Scenariul
   1. Descrierea scenei și a obiectelor

Am început realizarea proiectului prin construirea scenei, folosind aplicația Blender. Site-urile cu obiecte 3D, strict cu extensia .obj, m-au ajutat să pot înfrumuseța scena și să adaug obiecte după plac. M-am inspirat din atmosfera de curte chiar de la mine de acasă. Astfel, am încorporat în proiectul meu o casă, un felinar, câțiva brăduți, o minipiscină cu umbrelă de soare și minge de plajă, o mică parte de grădină unde se regăsesc trandafiri, o mașină lowpoly, un grill pentru grătare, o masă primitoare cu două scaune și două viețuitoare care să reîntregească peisajul: o pasăre și o pisică.

A computer screen shot of a house

Description automatically generated

3.2 Funcționalități

În crearea aplicației, am reușit să realizez următoarele:

* vizualizarea scenei: scalare, translație, rotație, mișcarea camerei
  + utilizând tastatura sau mouse-ul
  + utilizând animații de prezentare
* o scenă care dispune de sursă de lumină
* vizualizare scenă în modurile solid, wireframe, poligonal și smooth
* maparea texturilor și definirea materialelor
  + calitatea texturilor și nivelul de detaliu al acestora
  + maparea texturilor pe obiecte
* exemplificarea animării diferitelor componente ale obiectelor

Mașina, masa, scaunele, grătarul, trandafirii, felinarul și casa, nu au venit cu texturi gata mapate, ci am creeat eu pentru fiecare, la fel ca în tutoriale, texturi pe care mai apoi să le mapez. Totodată, și la realizarea planului mare, m-am folosit de textură pentru iarbă, am realizat câteva dealuri și am construit și o depresiune în care să fie apa. Mai jos, am atașat o imagine capturată a felului în care mapam texturile, similar am făcut și pentru celelalte obiecte.

A screenshot of a computer

Description automatically generated

A close-up of different colors

Description automatically generated

1. Detalii de implementare

După ce am rezolvat toate laboratoarele, în special laboratorul 10, pe codul acela am început să implementez scena mea, ștergând nanosuit-ul și componentele de care nu am avut nevoie, dar păstrând sursa de lumină, skydom-ul și shader-ele. Am adăugat majoritatea funcționalitățiilor în main.cpp si Camera.cpp. După cum se poate observa, îm primul rând am făcut posibilă vizionarea scenei în cele 4 moduri cerute, astfel:

A screenshot of a computer program

Description automatically generated

Tasta M – GL\_FILL, modul solid. Am atașat mai multe perspective ale scenei mai jos:

A screenshot of a computer

Description automatically generated A computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer game

Description automatically generated A computer screen shot of a house and a pool

Description automatically generated

Apăsând tasta B, avem modul GL\_POINT:

A screenshot of a computer

Description automatically generated

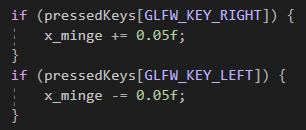
Tasta N, pentru modul GL\_LINE(wireframe)

A computer screen shot of a building

Description automatically generated

Din tastele J și L, se poate învârti stânga, respectiv dreapta, sursa de lumină. Dacă mișcăm mouse-ul, se mișcă și camera, iar din tastele Q și E, putem efectua rotații ale camerei.

În final, am reușit să mai implementez și o animație de prezentare a unei părți a conținutului scenei, care se poate activa cu tasta 1 si opri cu tasta 0 și am mai adăugat o minge de plaja care să aibă și ea la rândul ei o animație, aceea de a se mișca stânga/dreapta, chiar în funcție de tastele left și right. Atât scena, cât și mingea, au fost încărcate cu LoadModel în initObjects, la fel cum s-a făcut pentru lightCube și screenQuad. Mi-am setat și câteva settere și gettere in Camera.cpp, ca să pot lucra mai ușor și am creeat acolo funcția de prezentare. Mai jos am atașat și funcțiile pe care le-am folosit ca render pentru scena mea și separat pentru minge, care mai apoi au fost apelate în renderScene( ):

 A computer screen shot of a code

Description automatically generated

A screen shot of a computer code

Description automatically generated

A computer screen shot of a code

Description automatically generated

A computer screen shot of a program

Description automatically generated

A computer screen shot of a house with a dog and umbrellas

Description automatically generated

În ceea ce privește lumina, am folosit codul învățat la laborator, adică o lumină direcțională – ambientală, difuză și speculară. La rotire, se poate vedea că luminează anumite obiecte. Shadow mapping este o tehnică multi-trecere care utilizează texturi de adâncime pentru a decide dacă un punct se află în umbră sau nu. Cheia este aceea de a observa scena din punctul de vedere al sursei de lumină în loc de locația finală de vizionare (locația camerei). Orice parte a scenei care nu este direct observabilă din perspectiva luminii va fi în umbră.

1. Prezentarea interfeței grafice utilizator / manual de utilizare

Tasta Q – rotație în scenă dreapta Tasta J – rotație cub lumină, dreapta

Tasta E – rotație în scenă stânga Tasta L – rotație cub lumină, dtânga

Tasta W – deplasare înainte Tasta B – vizualizare GL\_POINT

Tasta S – deplasare înapoi Tasta N – vizualizare wireframe

Tasta A – deplasare stânga Tasta M - vizualizare smooth

Tasta D - deplasare dreapta Tasta Z – deplasare în sus

Tasta 1 – animație de prezentare Tasta X – deplasare jos

Săgeată stânga – deplasare stânga minge Săgeată dreapta – deplasare dreapta minge

1. Concluzii și dezvoltări ulterioare

Proiectul propus și realizat aduce în prim plan o reprezentare 3D captivantă și interactivă, folosind tehnologiile învățate de-a lungul semestrului. Odată cu îndeplinirea tuturor cerințelor, am obținut o scenă fotorealistă și dinamică, oferind utilizatorului posibilitatea de a explora și manipula mediul 3D într-un mod intuitiv. Documentația detaliată asigură o înțelegere completă a implementării și oferă resurse clare pentru utilizatori, facilitând interacțiunea și extinderea ulterioară a proiectului.

Consider că, proiectul meu ar putea avea o multitudine de feluri în care ar mai fi putut fi dezvoltat, de la ceață și umbre până la un adevărat joc. Însă, din cauza unei funcționalități nu tocmai bune a laptop-ului și nevoia de a aștepta chiar și 10 minute pentru ca OpenGL Example să apară, nu am reusit să îl extind cu mai multe obiecte, și așa mișcându-se foarte sacadat. Una peste alta, am descoperit încă o dată cât de interesantă poate fi programarea.

1. Referințe
   * [Tutoriale Blender - YouTube](https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM)
   * [Tutorial Blender - Documente Google](https://docs.google.com/document/d/1njtWPMmOQNIaD_z9ve8iPRUqQTWdIV_PO-NvPD0nOuM/edit#heading=h.7h28ckb0ku81)
   * [3D Models for Free - Free3D.com](https://free3d.com/)