Darius Razvan Radu

universitatea tehnica “cluj-napoca”

Proiect prelucrare grafica

Contents

[1. Prezentare temei 2](#_Toc124719145)

[2. Scenariu 2](#_Toc124719146)

[2.1. Descrierea scenei și a obiectelor 2](#_Toc124719147)

[2.2. Funcționalități 3](#_Toc124719148)

[3. Detalii de implementare 4](#_Toc124719149)

[3.1. Funcții și algortmi 4](#_Toc124719150)

[3.2. Modelul grafic 5](#_Toc124719151)

[3.3. Structuri de date 6](#_Toc124719152)

[3.4. Ierarhia de clase 6](#_Toc124719153)

[4. Prezentarea interfeței grafice utilizator / manual de utilizare 6](#_Toc124719154)

[5. Concluzii și dezvoltări ulterioare 7](#_Toc124719155)

[6. Referințe 7](#_Toc124719156)

[ Materialele din laborator 7](#_Toc124719157)

# Prezentare temei

Pentru a ne familiariza cu mediul de lucru OpenGL, cat si pentru a invata concepte de baza de grafica 3D, am avut de implementat o scena cat mai realista, complet functionala cu detalii si efecte/animatii cat mai diverse si complexe. Principalul scop al acestei teme este de a invata aprofunda conceptele de grafica invatate la laborator, dar si de a ne folosi imaginatia pentru a realiza scena cat mai realist posibil.

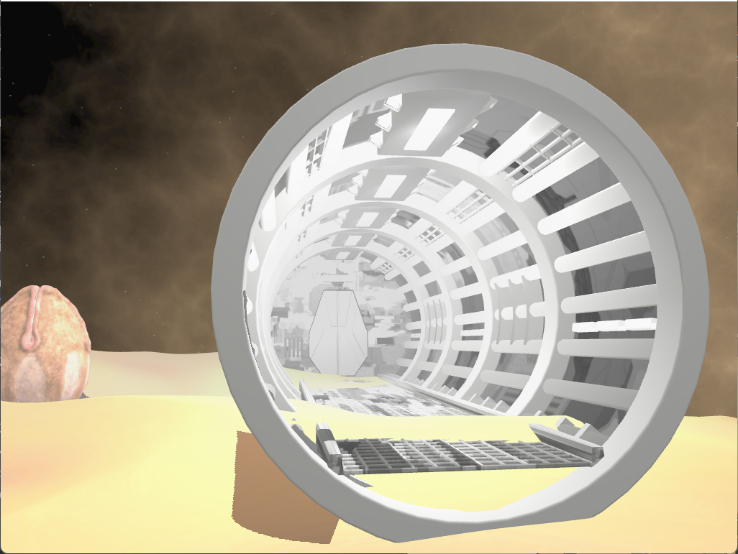
Acest proiect este o simulare de implementare a unui joc video simplist.

# Scenariu

## Descrierea scenei și a obiectelor

Scena descrie o planeta extraterestra. Scena este intr-un desert, plin de nave prabusite.

Am adugat o varientate de obiecte ce au legatura cu tema aleasa (spatiul cosmic si viata extraterestra).

Un dome, ce poate reprezenta asezarea unor oameni ce au populat planeta. Un coridor in care putem intra, ce va fi luminat, adaugand lumini punctiforme.

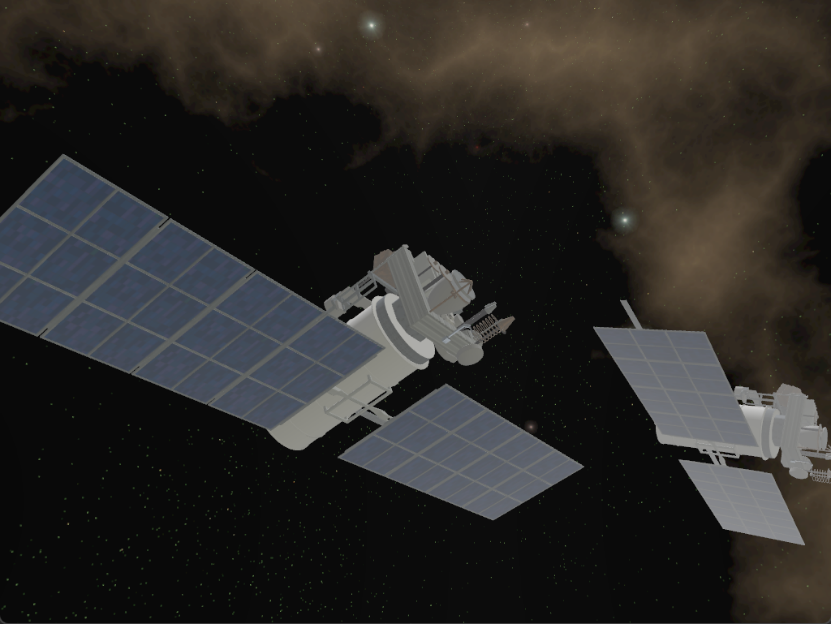
A picture containing building, dome

Description automatically generated

Pentru ca tema este spatiul cosmic, am putut adauga obiecte ce plutesc, deci am adaugat nave spatiale (acestea sunt animate pentru a se misca) si sateliti. Pentru ca tema nu imi permite a aduga multa vegetatie, din cauza mediului descris de tema, am adagat in loc de vegetatie oua de extraterestrii. De asemenea am adaugat obiecte texturate detaliat, 3 extraterestrii ce vor fi animati, putand da din cap.



A picture containing text, stationary

Description automatically generated

## Funcționalități

La functionalitati am implementat:

* Navigarea in scena cu ajutorul tastaturii si a mouse-ului.
* Zoom utilizand mouse-ul.
* Prezentarea automata a scenei la apasarea unui buton.
* Reflexia luminii pe obiecte.
* Umbrele obiectelor.
* Vizualizarea scenei in modurile solid, wireframe si pointframe.
* Animarea capetelor extraterestrilor.
* Animarea navelor spatiale.
* Crearea unui „laser” de lumina.
* Efect de fulger.
* Efect de ceata.
* Transparenta.

# Detalii de implementare

## Funcții și algortmi

* Am facut functii dedicate pentru randarea obiectelor ce trebuie animate. In aceste functii am modificat modelul obiectelor pentru a le translata pe anumite traiectorii. Pentru animatia capetelor si a navelor ce se misca am folosit o rotatie si o translatie simpla.
* Pentru efectul de laser, am importat 2 cilindre si 2 lumini punctiforme. Acestea se vor deplasa cu o viteza pe o linie dreapta dupa formula P = t \* P1 + (1-t)\*P0. Pentru a da un efect ca si cum nava ar trage de 2 ori, am translatat luminile pe rand, dand o iluzie de dinamism. Am schimbat random culoarea luminilor pentru a da un efect de curcubeu, pentru ca laserul sa arate mai bine.
* Functionalitatea de animatie: am realizat un algoritm in care camera se misca frame cu frame, pentru a da iluzia de miscare. Pentru inceput, am realizat miscarea din animatie plimbandu-ma in scena cu ajutorul tastaturii. In timpul miscarii, am printat intr-un fisier fiecare actiune a tastelor.

Odata ce am realizat fisierul pentru actiuni, am citit pe rand fiecare miscare si am mutat camera aferent, exact in modul in care se realizeaza cand apasam pe WASD.

De asemenea, pentru a prezenta fiecare efect/animatie, a trebuit sa ma opresc in anumite punct de control cateva frame-uri. Acest lucru a fost realizat cu ajutorul unei helperFunction.

Acest helper function va apela functia time() pentru a prelua timpul curent in secunde. Am stocat punctele de control cat si anumiti parametri utilizati pentru fiecare punct de control. Din functia de animatie, am apelat acest helper function pentru fiecare punct in care dorim sa ne oprim pentru a vizualiza animatia/efectul. Functia verifica daca ne aflam in punctul de control, daca ne aflam in acesta vom activa un flag care va randa aceasi pozitie pentru o persioada de timp, cat si flag-ul aferent animatiei/efectului dorit pentru a fi vizualizat. Perioada de timp o declaram noi pentru fiecare punct in parte. In momentul in care s-a terminat timpul (calculam acest lucru folosind functia time()), vom reporni animatia dezactivand flag-ul.

De asemenea, pentru

* Generarea de umbre a fost realizata ca in lab, folosind un shadowMap. Acest shadowMap a fost stocat intr-o textura folosind un EBO, iar apoi va fi aplicata in fragment shader peste scena pe care dorim sa o afisam.
* Am realizat transparenta, adaugand alpha ul din textura la culoarea finala in fragment shader.
* Am realizat ceata, folosind o functie de interpolare a culorii.

𝑓𝑜𝑔𝐹𝑎𝑐𝑡𝑜𝑟 = 𝑒 −(𝑓𝑟𝑎𝑔𝑚𝑒𝑛𝑡𝐷𝑖𝑠𝑡𝑎𝑛𝑐𝑒 ∗ 𝑓𝑜𝑔𝐷𝑒𝑛𝑠𝑖𝑡𝑦) 2

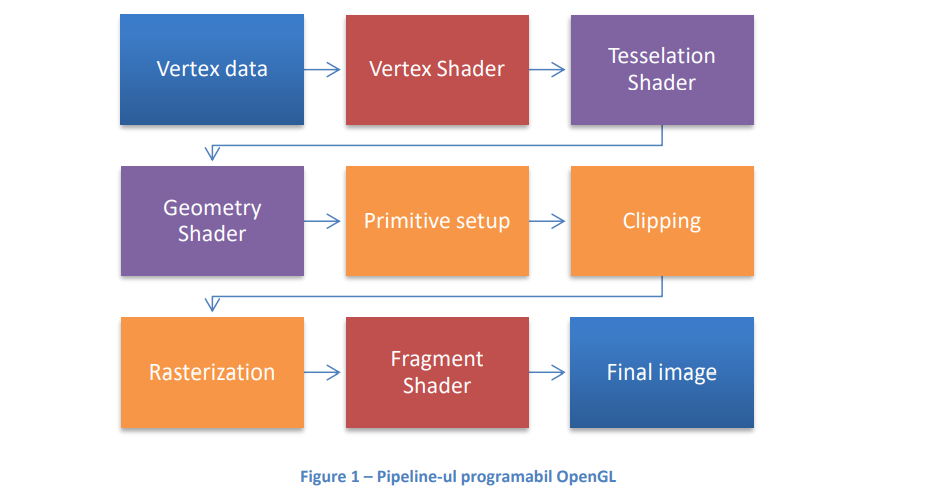
* In scena am adaugat fum, acest efect a fost realizat prin metoda de eliminare a fragmentelor. Acest lucru a fost realizat prin eliminarea unui varf in momentul in care are alpha ul mai mic de 0.1.

Text

Description automatically generated

## Modelul grafic

In OpenGL este ilustrat în figura de mai jos. Pipeline-ul programabil conține cel puțin stagiile de vertex shading și fragment shading, și înlocuiește pipeline-ul fix, care nu mai este suportat (iluminare și transformări). Faza de vertex shading procesează fiecare vârf independent de celelalte; datele vârfurilor sunt primite de la aplicație prin obiecte de tip vertex buffer (VBO). Faza de rasterizare generează un set de fragmente care sunt procesate în faza de fragment shading; ieșirea este reprezentată de culoarea și adâncimea fragmentului (coordonata Z). Unele etape (cum ar fi primitive setup, clipping și rasterization) nu sunt programabile. Tesselation Shader și Geometry Shader sunt opționale.



## Structuri de date

In afara de clasele deja existente in template-ul oferit in laborator si vectorii din biblioteca STL, am creat eu 2 structuri de date speciale pentru programul meu:

Pentru realizarea animatiei, am creat un enum care e folosit in momentele in care dorim sa ne oprim pentru vizualizarea animatiilor/efectelor. Acest enum este ca un flag, pentru ca noi avem 3 stadii in momentul animatiei. Stadiul de Init (cand inca nu am ajuns in punctul de control), stadiul de wait (momentul in care am ajuns si asteptam terminarea animatiei) si stadiul de done (momentul cand incepem din nou miscarea camerei)

Graphical user interface, text

Description automatically generated with medium confidence

De asemenea am realizat un struct ce contine punctele de control si parametrii necesari pentru asteptarea in punctul respectiv, cat si parametrii necesari pentru fiecare efect/animatie in parte

Graphical user interface, text

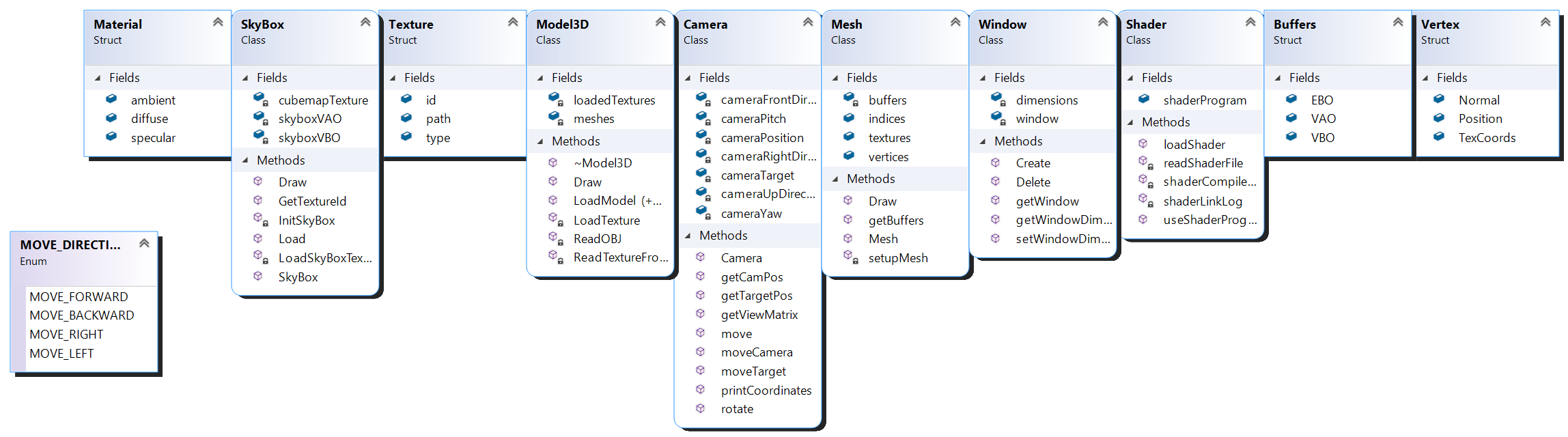
Description automatically generated

A doua structura este folosita pentru stocarea extraterestrilor. Pentru ca am realizat animatia fiecarui cap in parte, a fost nevoie exportarea fiecarui cap si corp ca si obiect separat.

Text

Description automatically generated

## Ierarhia de clase



# Prezentarea interfeței grafice utilizator / manual de utilizare

La rularea executabilului, va aparea o fereastra pe care se va prezenta scena descrisa mai sus. Pentru a beneficia de toate functionalitatiile pe care proiectul le ofera se vor folosi butoane de la tastatura.

Primul lucru pe care trebuie sa il stim, este ca putem naviga prin scena utilizand mouse-ul (sau sagetiile) si butoanele WASD. De asemenea vom putea da zoom folosind rotita mouse-ului.

Functiile pe care le ofera aplicatia se pot activa/dezactiva de la butoanele urmatoare:

* Tasta M: Desenarea pe ecran a hartii de umbre pentru a se exemplifica corectitudinea generarii de umbre.
* Tasta 0: Activarea animatiei in care navele se misca inainte.
* Tasta 9: Generarea efectului de lightning.
* Tasta 8: Activarea animatiei de laser.
* Tasta 7: Activarea animatiei.
* Tastele 6, 5, 4: Activarea animatiei pentru fiecare cap al extraterestrilor.
* Tastele J si L: Pentru mutarea luminii, pentru a exemplifica functionarea umbrelor.
* Tasta 1, 2, 3: Pentru schimbarea modului de vizualizare (solid, wireframe, pointframe).

# Concluzii și dezvoltări ulterioare

In viitor, pentru a se dezvolta aceasta aplicatie pot fi introduse mai multe efecte sau implementati algoritmi pentru a randa imaginile cat mai real posibil.

Exemple de functionalitati ulterioare:

Efect de bloom pentru lumini, introducerea de spotlights si colorarea conului de vizualizare al acestora, implementarea de coliziuni etc..

# Referințe

* Materialele din laborator
* <https://learnopengl.com/>