MINISTERUL EDUCAȚIEI ȘI CERCETĂRII ȘTIINȚIFICE

UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

Facultatea de Automatică și Calculatoare

Catedra: Prelucrare Grafică

Proiect final

Studentă: Strujan Florentina

Grupa: 302310

Semigrupa: 2

An: 2020-2021

Semestrul: 1

1.Cuprins

Cuprins

2.Prezentarea temei

Proiectele au ca și scop realizarea unei prezentări fotorealiste a unor scene de obiecte 3D utilizând librăriile prezentate la laborator (OpenGl, GLFW, GLM, etc.). Utilizatorul trebuie să aibă posibilitatea de a controla scena prin intermediul mausului și tastaturii.

(2p) vizualizarea scenei: scalare, translație, rotație, mișcarea camerei
-utilizând tastatura sau mausul
-utilizând animații de prezentare
(1p) specificarea surselor de lumina (cel puțin două surse de lumină diferite)
(0.5p) vizualizare scenă în modurile solid, wireframe, poligonal și smooth
(1p) maparea texturilor și definirea materialelor
-calitatea texturilor și nivelul de detaliu al acestora
-maparea texturilor pe obiecte
(1p) exemplificarea generării umbrelor
(0.5p) exemplificarea animării diferitelor componente ale obiectelor
(3p) fotorealism, complexitatea scenei, nivelul de detaliere al modelării, dezvoltarea diferiților algoritmi și implementarea acestora (generare dinamică de obiecte, detecția coliziunilor, generarea umbrelor, ceață, ploaie, vânt), calitatea animațiilor, utilizarea diferitelor surse de lumină (globală, locală, de tip spot)
(1p) documentația (obligatorie)

Tema acestui proiect este reprezentarea fotorealistică a unei scene care conține diferite obiecte 3D, folosind biblioteca OpenGL. Utilizatorul va putea interacționa cu scena cu ajutorul mouse-ului (deplasarea prin scena) sau a tastaturii (manipularea diferitelor obiecte).In acest proiect, a fost reprezentă o lume sf.

3.Scenariul

Rulând programul, se va deschide o fereastră conținând scena, împreună cu obiectele și efectele corespunzătoare.

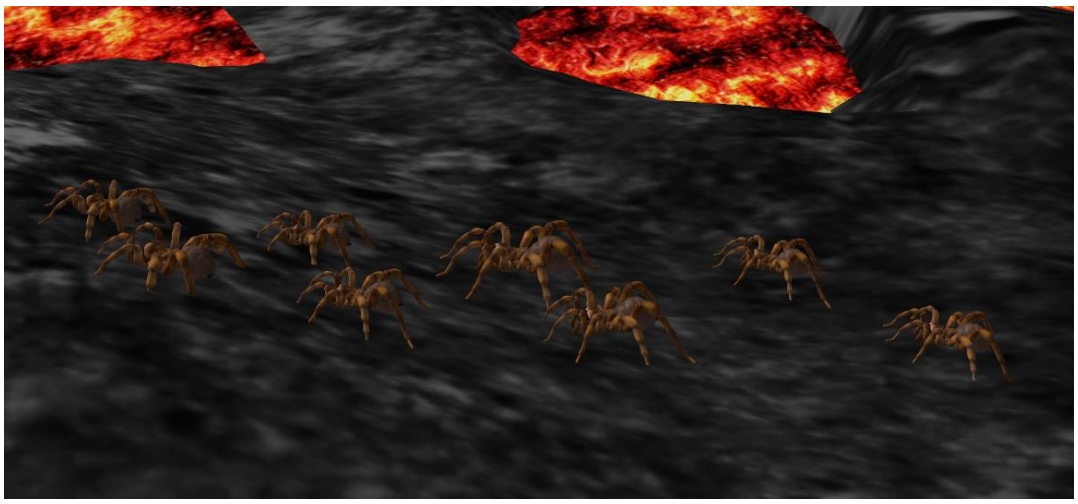
3.1.Descrierea scenei și a obiectelor

Scena este reprezentativă unei lumi SF, cu tot felul de monștri, dar și cu posibilitatea oamenilor de a ajunge în aceasta, deși, după cum se observă, cu 0 șanse de a se întoarce. Utilizatorul se poate plimba prin scenă cu ajutorul mouse-ului și a tastaturii, în rolul unui personaj al scenei și poate vizualiza scena în modul de prezentare Solid, Wireframe sau Point.

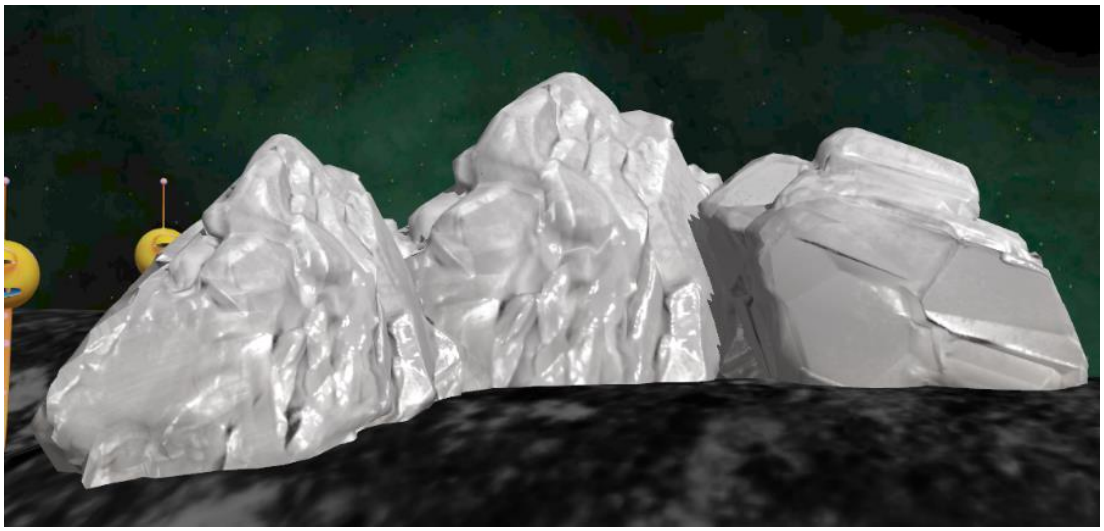
Obiectele folosite sunt:

- Un extraterestru

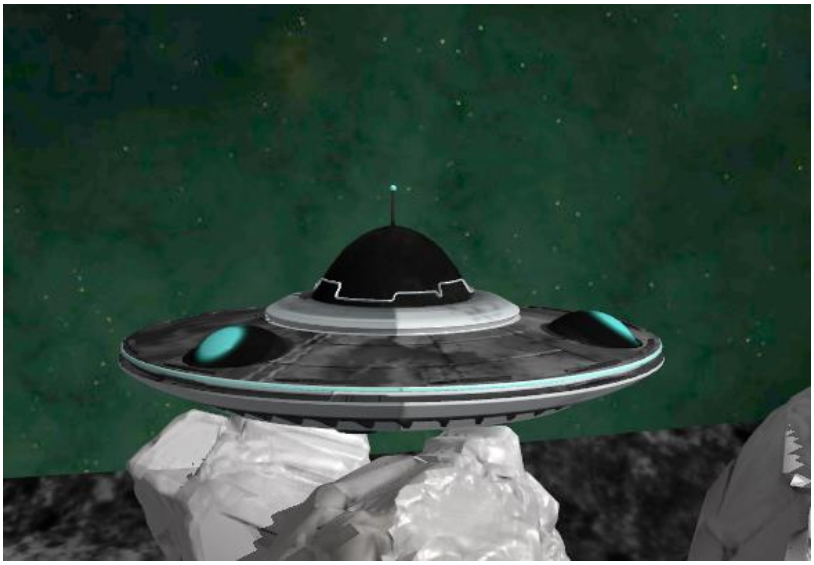
- Păianjeni



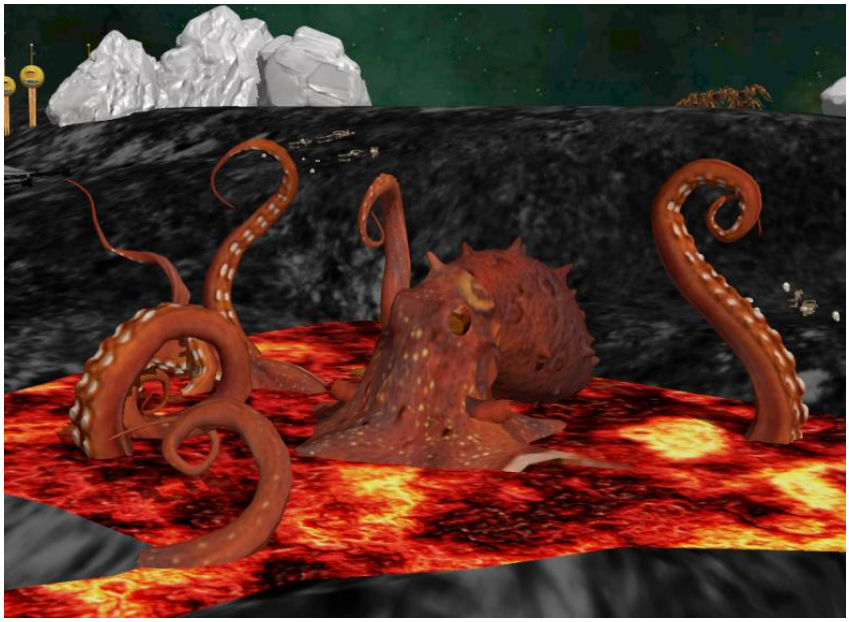
- Stânci



- Ozn-uri

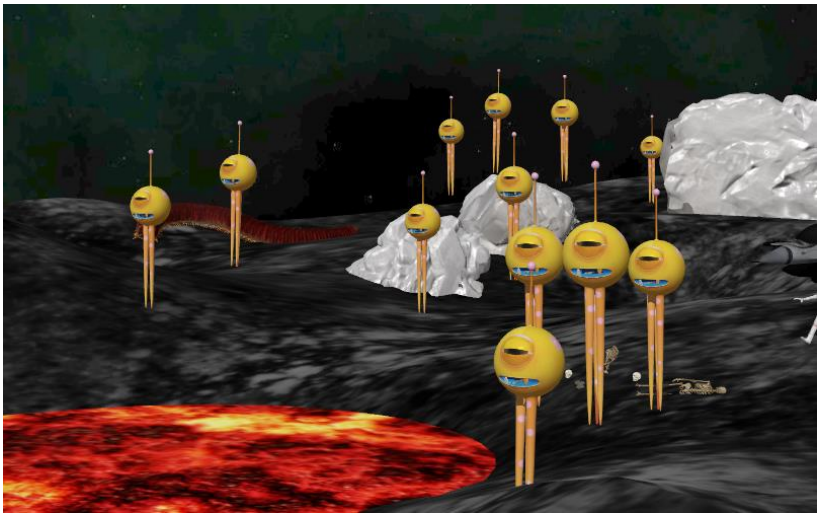


- Caracatițe de lavă

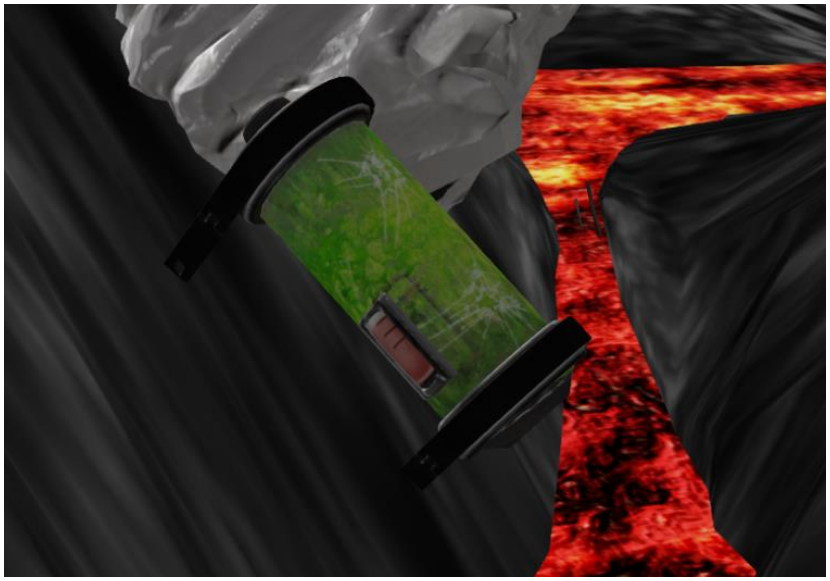


- Un păzitor (moartea cu coasa)

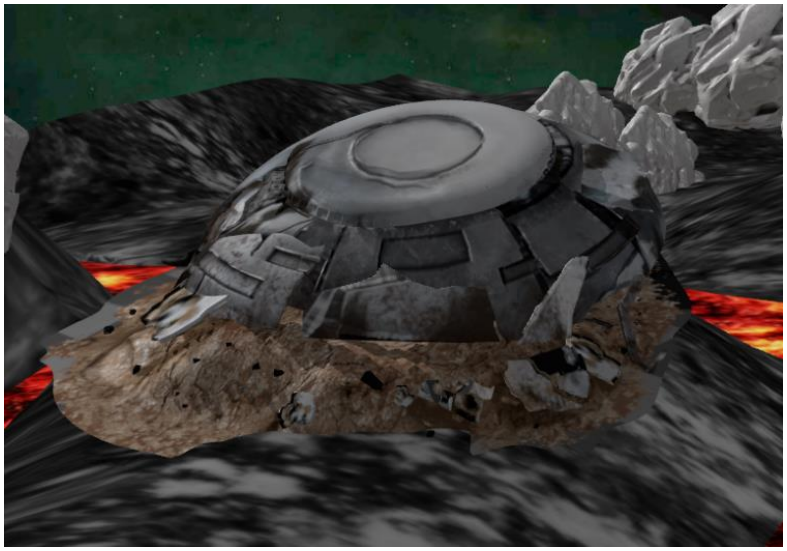
- Monstruleți galbeni



- Recipient cu substanță verde

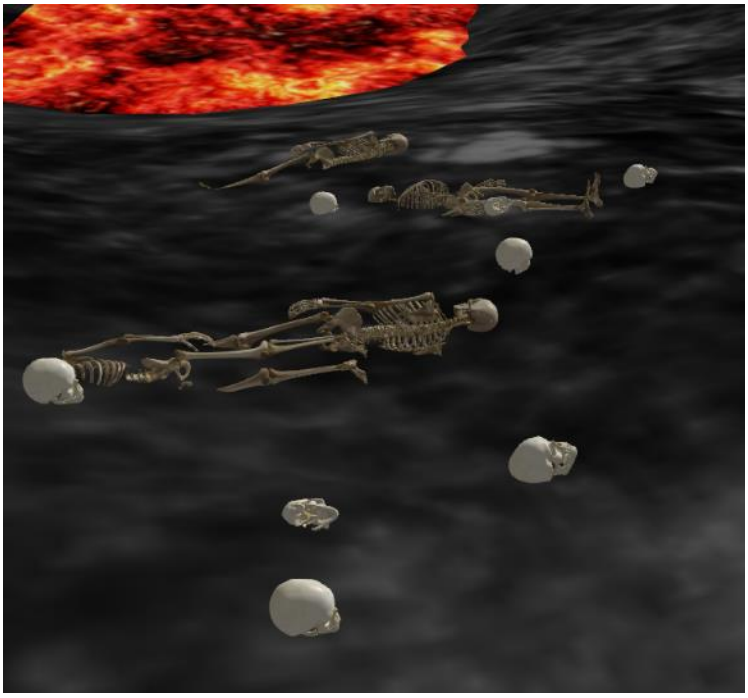


- O navă prăbușită

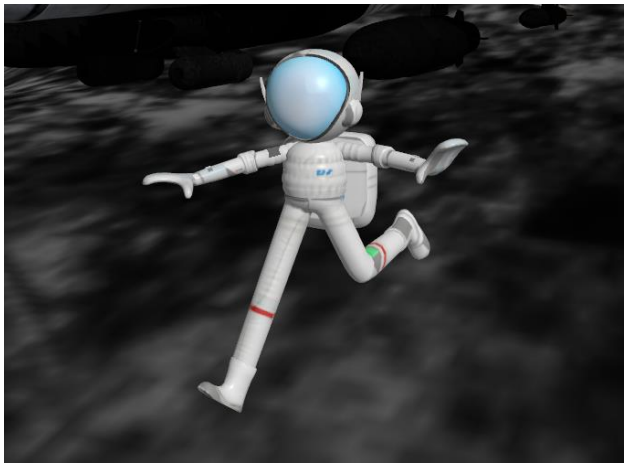


- Viermi

- Schelete și cranii



- Un astronaut și racheta cu care a aterizat



3.2.Funcționalități

Utilizatorul se poate deplasa prin scena cu ajutorul tastelor (W,A,S,D) precum si prin mișcarea mouse-ului. Scena are o sursă de lumină direcțională care poate fi deplasată în jurul scenei cu ajutorul tastelor Z și X. Sursa de lumină punctiformă poate fi pornită sau oprită folosind tastele K și L. Aceasta i se atribuie unui recipient cu o substanță verde. Vremea poate fi schimbată prin apariția ceții folosind tasta P și dispariția ei cu tasta O. Momentul zilei poate fi schimbat prin apariția întunericului folosind tasta U și prin revenirea la zi prin tasta I. Obiectele de tip nave spațiale își fac apariția în scenă prin tasta N. Prin apăsarea tastei G se schimbă modul de vizualizare a scenei(line/point/smooth).

4. Detalii de implementare

4.1. Funcții și algortmi

Libraria OpenGL are o larga varietate de functii cum ar fi glfwCreateWindow() cu ajutorul careia putem crea o fereasta pe care ne vom proiecta scena, glGenTexture(), glBindTexture(),glTextImage2D() cu ajutorul caruia putem sa creeam texture pentru obiecte, si multe alte functii.

Una dintre cele mai esentiale functii din acest proiecte este renderScene(), care este folosita pentru a trimite toate datele la “shadere” si a calcula diferite valori necesare proiectarii scenei. In aceasta functie sunt create toate modele cu umbrele lor, locul in care sunt apelate diferite functii care sa animeze obiecte (translatie, rotatie) , precum si simpla pozitionare a acestora in scena printr-o miscare de translatie. O alta functie importanta este initUniforms() unde sunt calculate valori pentru reprezentarea iluminarii (directia luminii, culoarea etc). Functia initShaders() este o functie in care sunt instantiate toate shaderele, precum si functia initObjects() in care sunt de asemenea instantiate toate modelele 3D (obiectele din scena). Cu ajutorul functiilor processMovement(), move(), mouseCallBack(), s-a putut realiza diferitele manipulari asupra modelelor 3D (miscarea anumitor obiecte), sau asupra camerei (miscarea prin scena).

A fost folosit un skybox pentru a reprezenta peisajele indepartate de scena (cerul).

4.1.1. Soluții posibile

Pentru aceasta aplicatie pot fi imbunatatite texturile unor obiecte, folosirea mai multor surse de lumina, crearea de animații pentru obiecte, dezvoltarea unui numar mai mare de obiecte.

4.1.2. Motivarea abordării alese

Am ales acest mod de implementare datorită resurselor de la laborator și a ușurinței de executare, încercând și alte moduri mai interesante, dar care nu au reușit într-un final. Abordarea aleasa se pliaza pe laboratoarele parcurse si informatiile folosite la acestea. Functiile folosite la laborator sunt mai mult decat sufficiente pentru a crea o aplicatie. Acestea totusi se pot dezvolta ulterior.

4.2. Modelul grafic

Obiectele din scena sunt Modele 3D, cu extensia .obj care au fost preluate de pe o aplicație numită 3D Viewer, initial cu extensia .gltf, urmand sa separ textura de obiect in blender pentru a le putea importa pentru Visual Studio.

Modelul grafic utilizat este modelul prezentat in lucrarile de laborator ,model ce se bazeaza pe modelul ShadowMapping, pe modelul Phong, un model empiric al iluminarii punctelor pe o suprafata folosind componentele luminii: difuza, ambientala, speculara.

4.3. Structuri de date

Pentru realizarea calculelor am folosit structurile de date disponibile in biblioteca GLM, cum ar fi vec<n>, sau mat<n>, precum si structurile specifice OpenGL-ului cum ar fi Gluint, GLFWwindow, etc.

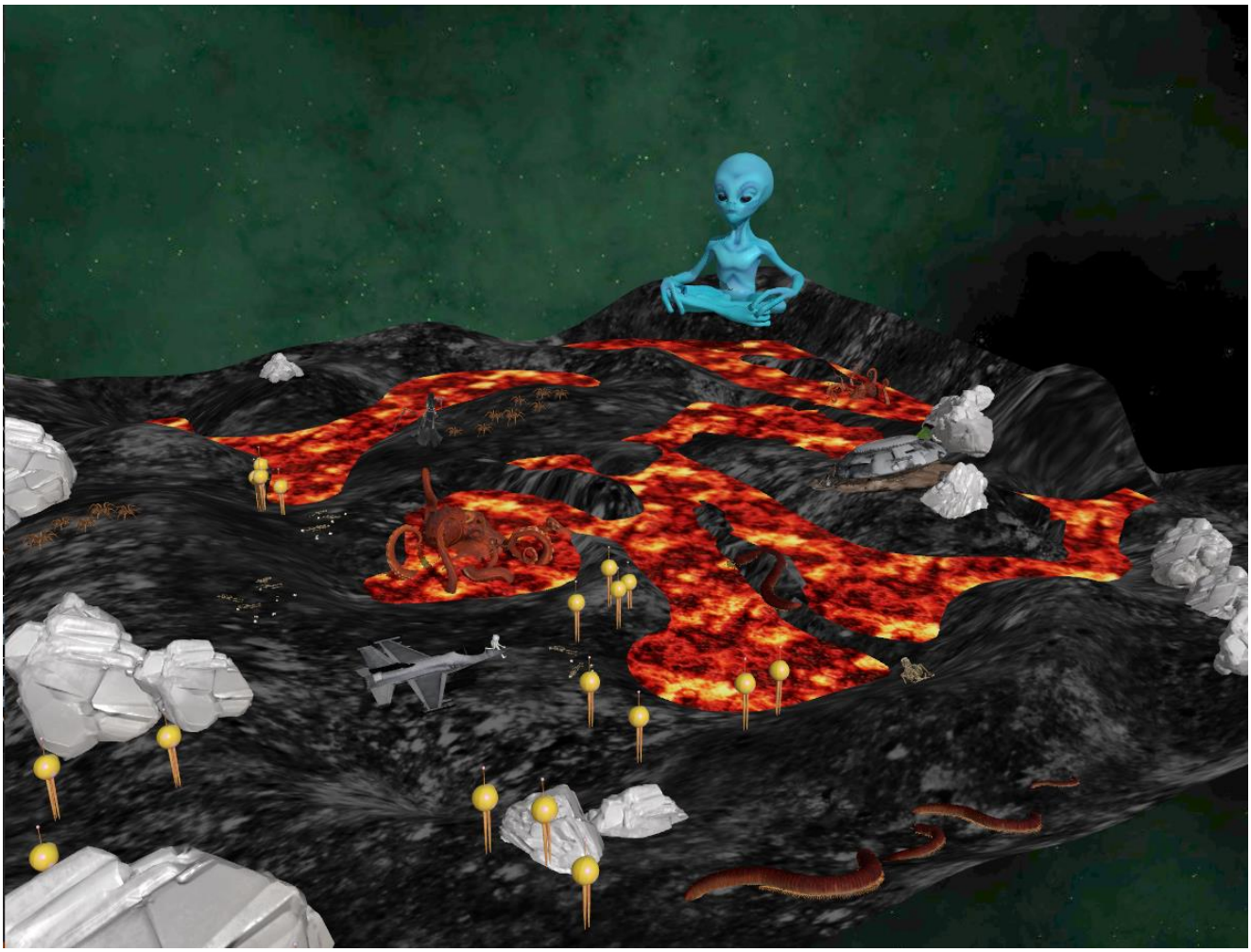
4.4. Ierarhia de clase

Ierarhia de clase a proiectului este definite prin diferitele librarii. De exemplu Mesh.cpp and Models3D.cpp sunt folosite definirea obiectelor precum si pentru aplicarea de texture asupra acestora. Cu ajutorul Shader.cpp se vor putea compila si lega shaderele existente de program. Camera.cpp este locul in care sunt definite functii ce se aplica asupra camerei (miscarea acesteia).

Pentru a implementa functionalitatile implementate anterior, a fost necesara includerea de headere si .cpp-uri precum:

- Camera.hpp – care realizeaza controlul camerei
- Mesh.hpp – defineste varfurile unui obiect
- Model3D.hpp – creeaza un nou model 3D
- Shader.hpp – incarca un shader
- SkyBox.hpp – realizeaza functionalitatea cubemap-ului
- glm.hpp – biblioteca pentru calculul matematic
- GLEW.h si GLFW.h – pentru functionalitate/randari

5. Prezentarea interfeței grafice utilizator / manual de utilizare

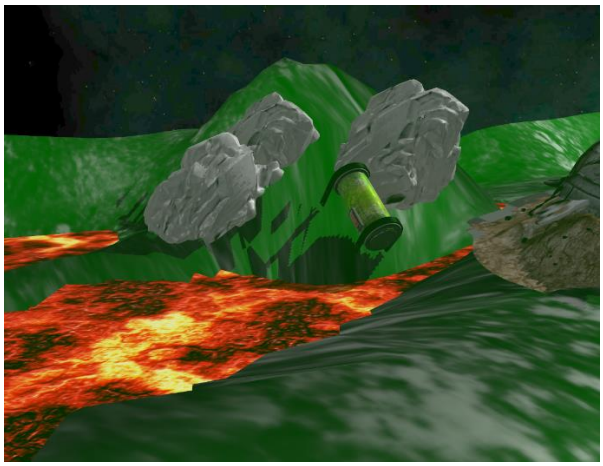


Manual de utilizare:

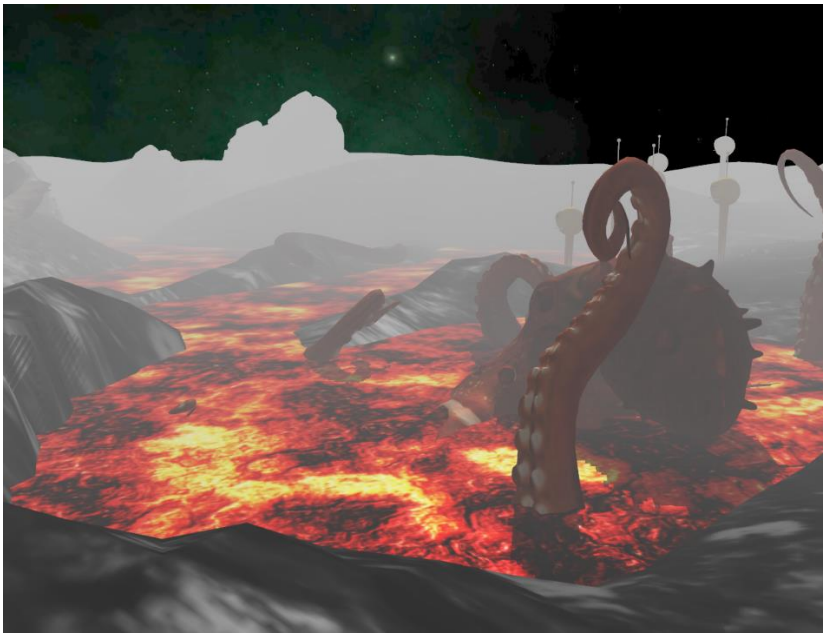
- “W” -mișcare înainte
- “A” - mișcare înapoi
- “S” - mișcare la stânga
- “D” - mișcare la dreapta
- “mouse” - rotație camera

- “Q” - rotație scenă spre stânga
- “E” - rotație scenă spre dreapta
- “Z” - mișcare lumină direcțională la stânga
- “X” - mișcare lumină direcțională la dreapta

- “K”- pornire sursă de lumină punctiformă
- “L”-oprire sursă de lumină punctiformă



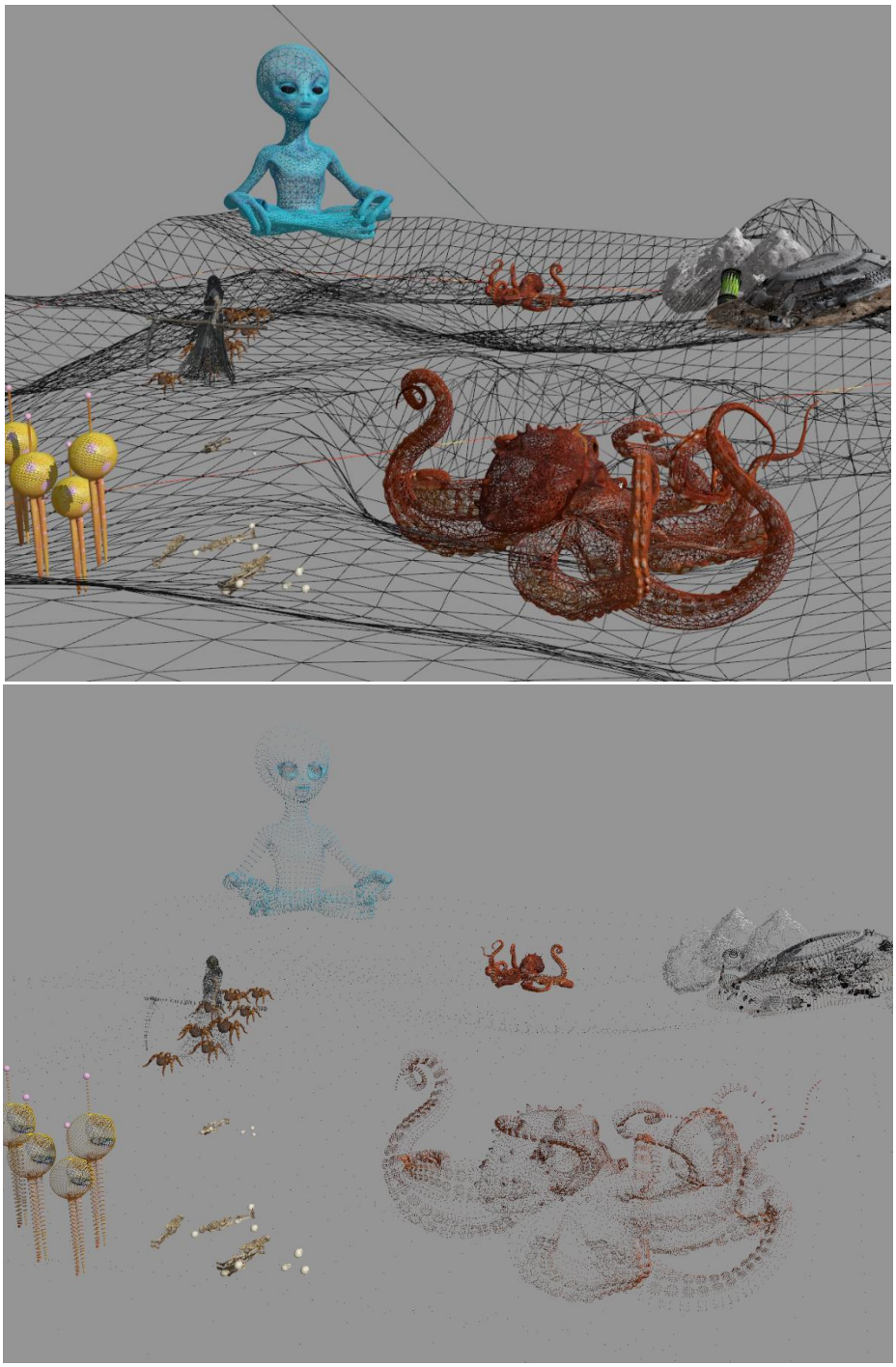
- “P”- pornire ceață
- “O”-oprire ceață



- "U"- începe noapte
- "I"-revenire la zi

- "N"- apariție nave

- "G"-schimbare mod vizualizare line/point/smooth



6. Concluzii și dezvoltări ulterioare

Pot spune ca, prin crearea aceste scene, mi-am imbunatait semnificativ skill-urile de blender, modelare 3D, openGL. Ca implementari ulterioare, s-ar putea adauga detectarea coliziunilor, animatii diverse si o imbunatatire a calitatii umbrelor.

Consider ca optimizarea memoriei poate fi realizata in urma aprofundarii cunostiintelor in acest domeniu.

7. Referințe

Îndrumătorul de laborator