

Prelucrare grafica - Proiect

Student: Bakk Cosmin-Robert

Grupa: 30237

Cuprins

Prezentarea temei.....	2
Scenariul.....	3
Descrierea scenei si a obiectelor	3
Functionalitati	7
Detalii de implementare	9
Functii si algoritmi.....	9
Solutii posibile.....	9
Motivul abordarii alese	9
Modelul grafic	9
Structuri de date	9
Ierarhia de clase	9
Prezentarea interfetei grafice utilizator / manual de utilizare	10
Concluzii si dezvoltari ulterioare	12
Referinte	13

Prezentarea temei

Proiectul are 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 mousei și tastaturii.

Cerintele proiectului sunt următoarele:

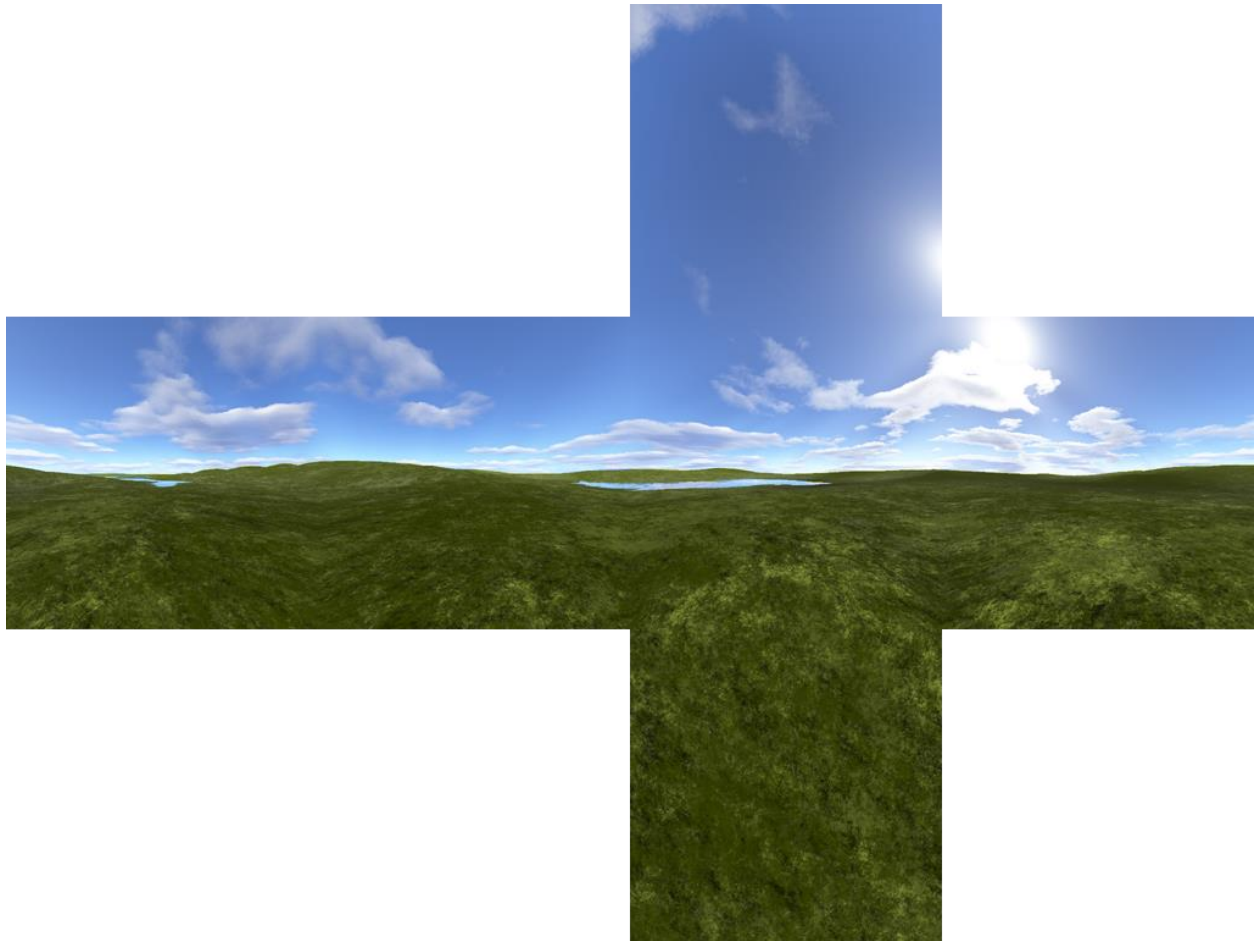
- vizualizarea scenei: scalare, translație, rotație, mișcarea camerei
 - utilizând tastatura sau mouseul
 - utilizând animații de prezentare
- specificarea surselor de lumina (cel puțin două surse de lumină diferite)
- vizualizare scenă în modurile solid, wireframe, poligonal și smooth
- maparea texturilor și definirea materialelor
 - calitatea texturilor și nivelul de detaliu al acestora
 - maparea texturilor pe obiecte
- exemplificarea generării umbrelor
- exemplificarea animării diferitelor componente ale obiectelor
- 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)
- documentația (obligatorie)

Scenariul

La rularea executabilului .exe, utilizatorului i se va deschide o fereastră de dimensiuni 1900x950, în care se poate deplasa cu ajutorul mouse-ului și a tastaturii pentru a explora o scenă 3D care conține obiecte și diferite animații.

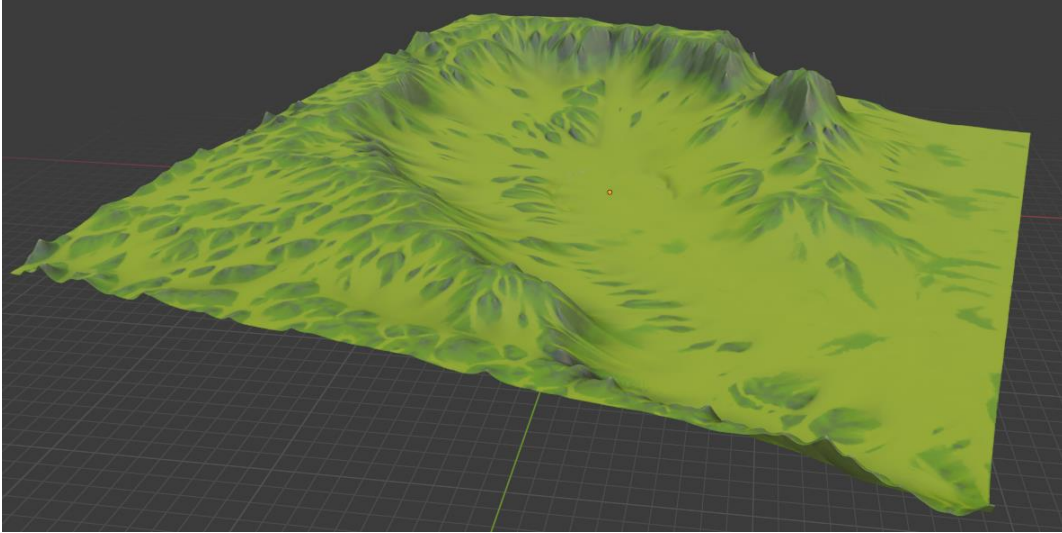
Descrierea scenei și a obiectelor

Scena este plasată în centrul unui skybox, care reprezintă o zonă de dealuri.

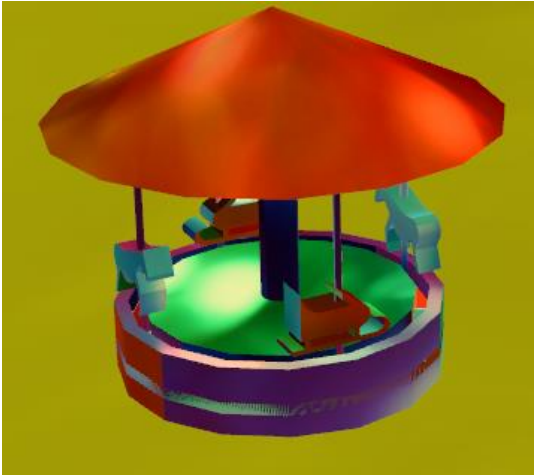


Scena contine:

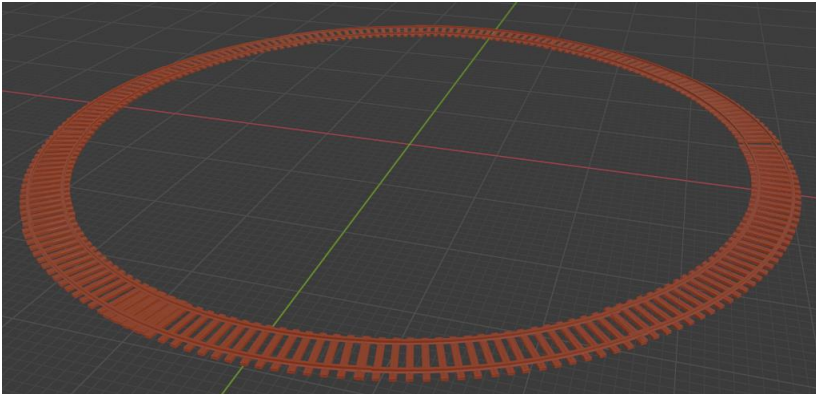
- un teren pe care sunt asezate obiectele



- un carusel, care se rotește în jurul centrului său mereu



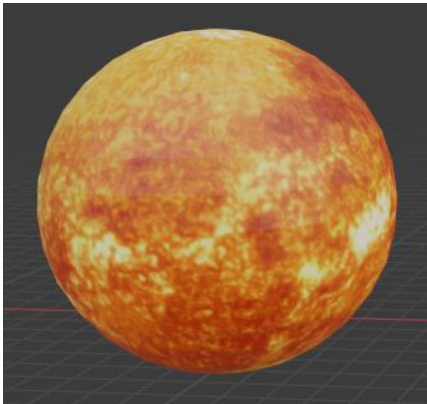
- o șină de tren, pe care va fi așezat trenul



- un tren



- o sfera de culoare portocalie in directia careia se afla o lumina directionala portocalie



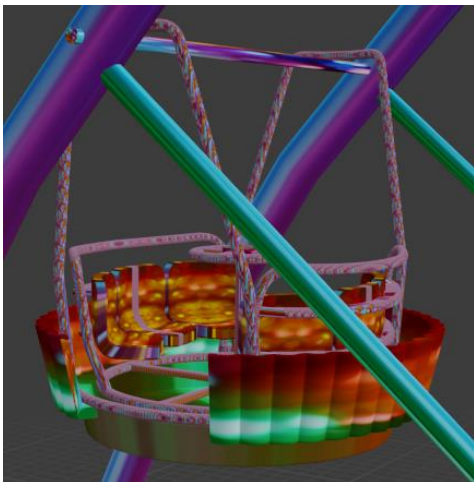
- o sfera de culoare alba in directia careia se afla o lumina directionala alba



- o roata care e formata dintr-o parte imobila si o parte mobila



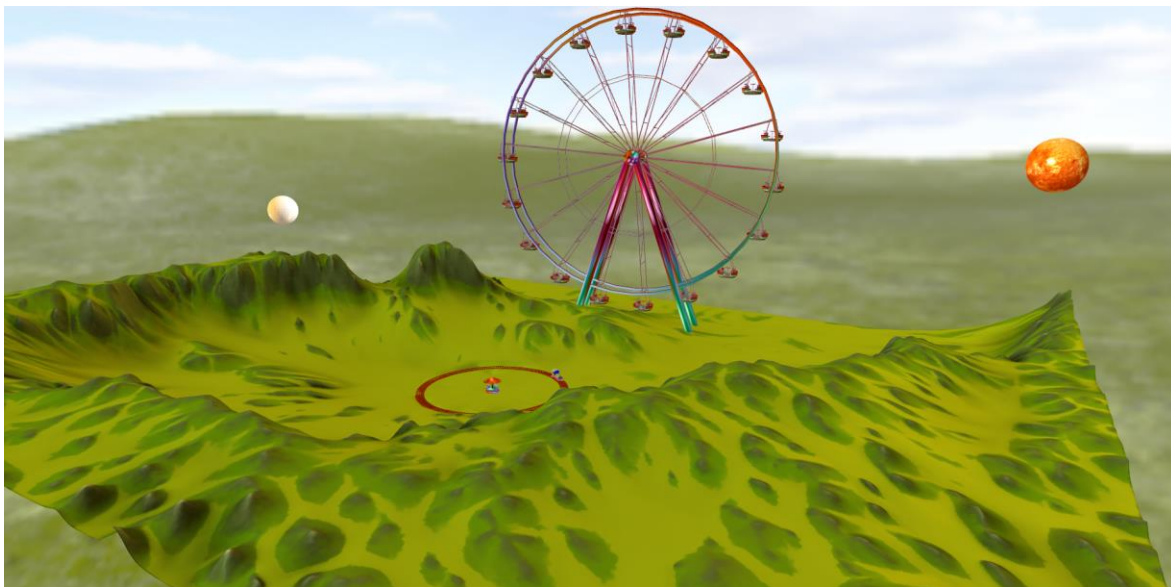
- o cabina care a fost translata in 18 pozitii pentru a fi pozitionata pe roata



- un text, "Train crashed!"



Scena completa:



Functionalitati

Caruselul se invarte incontinuu in jurul centrului sau.

Roata se invarte in jurul centrului sau. De asemenea, cabinele rotii se invart in jurul centrului rotii si se invart in acelasi timp in sens invers in jurul centrelor lor, pentru a ramane in pozitia initiala in timpul deplasarii in roata.

Utilizatorul se poate plimba prin scena cu ajutorul tastaturii si poate schimba directia de orientare a camerei cu ajutorul mouse-ului.

Utilizatorul poate roti sfera alba pe directie orizontala, schimbând astfel si directia din care porneste lumina de culoare alba.

Utilizatorul poate roti sfera portocalie pe directie verticala, schimbând astfel si directia din care porneste lumina de culoare portocalie.

Utilizatorul poate roti trenul pe sina tinând o tasta apasata, viteza trenului crescând cu cât este tinuta tasta apasata mai mult. De asemenea, trenul poate sari la apasarea unei taste, dar dacă sare în timp ce e în mișcare, va cădea de pe sina și va apărea textul "Train crashed!" deasupra sinei. Poziția trenului pe sina se poate reseta la apasarea unei taste.

Utilizatorul poate începe o animație de prezentare, în timpul căreia nu se mai poate mișca prin scena sau să își schimbe direcția de vizualizare. Animația de prezentare poate fi oprită, iar poziția camerei va rămâne în locul în care s-a terminat animația.

Utilizatorul poate crește sau descrește intensitatea cetei din scena, ceața fiind aplicată atât pe scena de obiecte, cât și pe skybox.

Utilizatorul poate privi scena în modurile solid, wireframe și point.

Detalii de implementare

Functii si algoritmi

Solutii posibile

Acest proiect putea fi implementat in moduri diferite, de exemplu obiectele puteau fi texturate in interiorul programului cu ajutorul comenzii `glBindTexture` sau texturate intr-un alt program, iar apoi incarcate in program. Skybox-ul putea sa fie de mai multe tipuri, cubemap sau skydome si sa se comporte in mai multe moduri, in functie de ce se dorea.

Motivul abordarii alese

Am ales sa lucrez cu obiectele si cu texturile in programul Blender, iar apoi sa incarc obiectele texturate cu functia `loadModel`. De asemenea, skybox-ul este de tip cubemap, caruia i-am asociat un shader propriu in care nu se aplica transformarea model, pentru a se comporta ca un obiect aflat foarte departe.

Modelul grafic

Modelul grafic folosit este cel prezentat in laboratorul de Prelucrare Grafica, bazat pe modelul Phong de colorare a obiectelor in functie de distanta acestora fata de lumina, folosind cele trei componente ale luminii: lumina difuza, ambientala si speculara.

Structuri de date

S-au folosit structurile de date disponibile in biblioteca GLM, cum ar fi `vecn`, `matn`, etc. si structurile disponibile in bibliotecile GLEW si GLFW, cum ar fi `GLuint`, `GLfloat`, etc.

Ierarhia de clase

Pentru implementare am folosit urmatoarele headere si .cpp-uri:

- Camera.hpp, Camera.cpp – pentru controlul camerei
- Mesh.hpp, Mesh.cpp – pentru desenarea obiectelor
- Model3D.hpp, Model3D.cpp – pentru definirea modelelor 3d
- Shader.hpp, Shader.cpp – pentru incarcarea si compilarea shaderelor
- Skybox.hpp, Skybox.cpp – pentru skybox-ul proiectului
- stb_image.h, stb_image.cpp – pentru incarcarea texturilor
- tiny_obj_loader.h, tiny_obj_loader.cpp – pentru incarcarea obiectelor
- Window.h, Window.cpp – pentru initializarea si lucrul cu fereastra
- Bibliotecile glm, GL si GLFW – pentru diferite functii OpenGL si glm

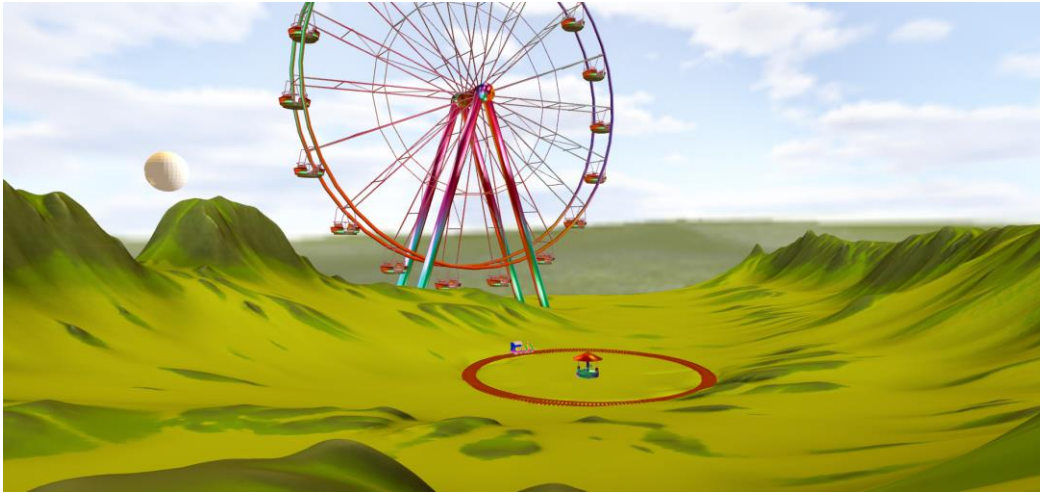
Prezentarea interfetei grafice utilizator / manual de utilizare

Se pot folosi urmatoarele taste pentru a realiza diferite actiuni in scena:

- “w” – deplasare in fata
- “a” – deplasare catre stanga
- “s” – deplasare in spate
- “d” – deplasare in dreapta
- rotirea mouse-ului – rotirea camerei
- “q” – trenul se roteste pe sina in timp ce aceasta tasta e apasata, viteza acestuia crescand in timp
- “e” – trenul sare
- “q” + “e” – trenul sare de pe sina si apare textul “Train crashed!”, dupa care trenul nu se mai poate roti sau sari pana cand se apasa tasta “r”
- “r” – trenul este resetat, iar textul “Train crashed!” dispare
- “v” + “1” – pornirea animatiei de prezentare
- “v” + “2” – oprirea animatiei de prezentare
- “h”, “j” – rotirea sferei albe si a luminii albe pe orizontala
- “k”, “l” – rotirea sferei portocalii si a luminii portocalii pe verticala
- “n”, “m” – decrementarea, respectiv incrementarea densitatii cetii, atat pe scena cat si pe skybox



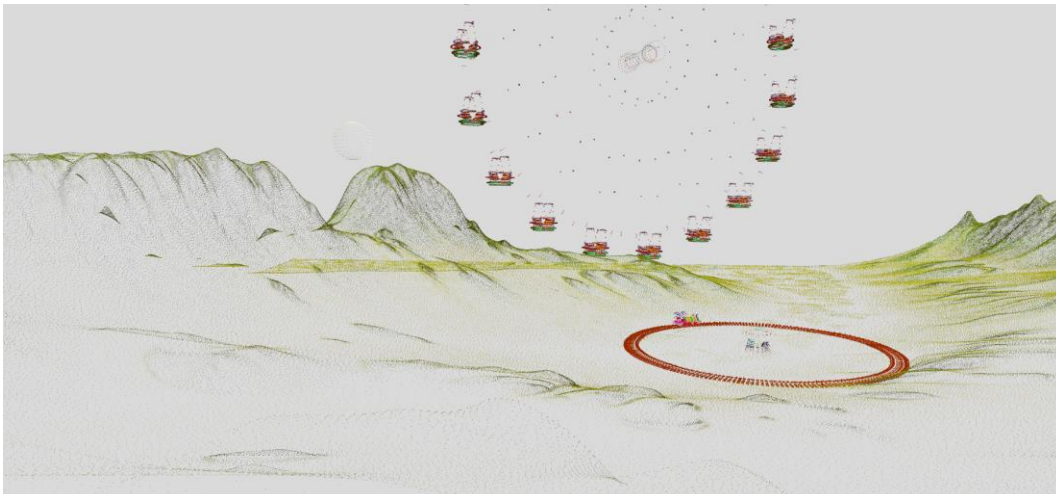
- “x” + “1” – vizualizarea scenei in modul solid



- “x” + “2” – vizualizarea scenei in modul wireframe



- “x” + “3” – vizualizarea scenei in modul point



Concluzii si dezvoltari ulterioare

Acest proiect m-a ajutat sa imi extind cunostiintele in limbajul C++, sa folosesc functionalitati ale librariilor OpenGL, sa modelez diferite obiecte 3D si sa le mapez texturi cu ajutorul programului Blender. Ca si dezvoltari ulterioare, in acest proiect s-ar putea implementa umbrele folosind ShadowMapping. De asemenea, s-ar mai putea adauga surse de lumina diferite, mai multe obiecte sau s-ar putea imbunatati texturile obiectelor din scena. Pentru mai mult realism, se pot adauga efecte de ceata/ninsoare.

Referinte

- [\(2\) Tutoriale Blender - YouTube](#)
- [Tiovivo 3D model | CGTrader](#)
- [Wheel Free 3D Models download - Free3D](#)
- [Tutorial Blender - Documente Google](#)
- Laborator Prelucrare Grafica