

PRELUCRARE GRAFICĂ

-PROIECT-

-SCENĂ INSULĂ EXOTICĂ-

Matiș Oana-Antonia

Grupa: 30238

Cuprins

1. Prezentarea temei.....	3
2. Scenariul	
2.1. Descrierea scenei și a obiectelor.....	4
2.2. Funcționalități	5
3. Detalii de implementare	
3.1. Funcții și algoritmi.....	6
3.1.1. Soluții posibile	9
3.1.2. Motivarea abordării alese	9
3.2. Modelul grafic.....	9
3.3. Structuri de date.....	9
3.4. Ierarhia de clase.....	10
4. Prezentarea interfeței grafice utilizator/manual de utilizare.....	11
5. Concluzii și dezvoltări ulterioare.....	12
6. Referințe.....	13

1. Prezentarea temei

Tema acestui proiect vizează prezentarea unei insule exotice aflate pe mare, care îndeplinește simultan rolul de resort de lux. Scopul principal al proiectului constă în crearea unei reprezentări 3D cât mai realiste și apropiate de detaliile vieții de zi cu zi din această locație paradisiacă.

Scena va permite vizualizarea insulei în timpul zilei, cu lumina soarelui reflectată în apa mării, dar în același timp va permite vizualizarea insulei în condiții mai puțin favorabile precum ceața.

Proiectul se axează pe aplicarea principiilor OpenGL și pe familiarizarea practică cu mediul de programare Visual Studio, implicând utilizarea unor librării precum OpenGL, GLFW, GLM etc. De asemenea, se va folosi mediul de modelare 3D Blender pentru a crea detaliile insulei și ale facilităților.

Utilizatorul va avea posibilitatea de a explora scena prin intermediul mouse-ului și tastaturii, permițându-i să navigheze în jurul insulei, să observe detaliile arhitecturale și să se bucure de atmosfera unică oferită de acest resort excepțional.



2. Scenariul

2.1. Descrierea scenei și obiectelor

Scena a fost creată cu ajutorul software-ului grafic "Blender" și include modele 3D cu extensia ".obj" preluate de pe diverse site-uri specializate în furnizarea de astfel de resurse. Aceasta reprezintă, de fapt, o insulă privată, exotică și de lux.

Insula este alcătuită dintr-un număr mediu de obiecte, care sunt dispuse într-un mod organizat pentru a crea atmosfera unei destinații exotice de vacanță. Pe insulă se regăsesc plaje cu nisip fin, un resort elegant, un yacht personal, facilități recreative, precum și elemente de arhitectură specifică insulelor tropicale. Scena cuprinde și detalii precum palmieri, zone de promenadă, dar și transportul privat oferit de resortul de lux de pe insulă.

Scena include și un loc special amenajat cu șezlonguri și o porțiune mai stâncoasă de unde se pot admira valurile mării.

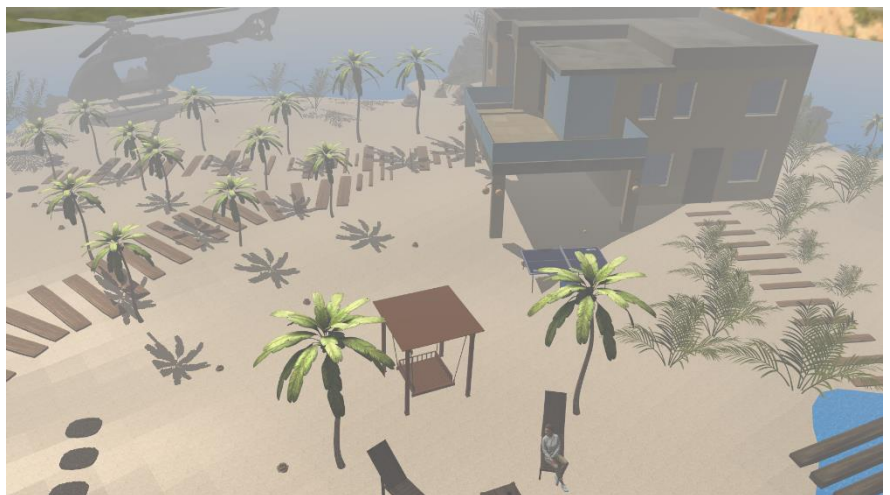


2.2. Funcționalități

În cadrul acestei scene, se oferă mai multe opțiuni de navigare, inclusiv navigarea prin intermediul mouse-ului și tastaturii. Pentru a accentua nivelul de realism, au fost implementate funcționalități cheie, cum ar fi lumini directionale și punctiforme, împreună cu seturi de shadere pentru obiecte, crearea hărții de adâncime pentru generarea umbrelor, gestionarea SkyBox și altele. În plus, pentru a adăuga dinamism scenei, s-a inclus o animație cu o bilă de ping-pong care se deplasează sus și jos și o scândură care plutește pe mare.

Funcționalitățile interactive sunt extinse și pot fi controlate prin tastatură, oferind utilizatorului posibilitatea de a iniția și opri diverse aspecte pentru a îmbunătăți experiența. De exemplu, se poate controla creșterea și scăderea în timp a unei cete, creând astfel iluzia unei lăsări și ridicări a acesteia. De asemenea, utilizatorul poate modifica intensitatea luminilor corpurilor de iluminare stradală și poate trece de la zi la noapte și invers.

Prin apăsarea diferitelor butoane ale tastaturii, utilizatorul poate vizualiza scena în mai multe moduri, inclusiv în mod solid, wireframe și punctiform, oferind astfel o varietate de perspective pentru explorarea și aprecierea detaliilor scenei 3D.



3. Detalii de implementare

3.1. Funcții și algoritmi

În mod inițial, avem funcțiile implementate pe parcursul laboratorului, enumerate astfel: *windowResizeCallback*, *keyboardCallback*, *mouseCallback*, *processMovement*, *initOpenGLWindow*, *initOpenGLState*, *initObjects*, *initShaders*, *initUniforms*, *initFBO*, *computeLightSpaceTrMatrix*, *drawObjects*, *renderScene*, și *cleanup*.



- **Ceața**

Efectul de ceață a fost abordat în laborator prin propunerea mai multor tipuri de ceață: ceață liniară, ceață exponențială și ceață exponențială pătratică. În cadrul acestui proiect, am ales să implementăm ceață exponențială pătratică. Calculul factorului de ceață este realizat în funcția "computeFog()" din shader-ul de fragmente, utilizând formula:

$$fogfactor = e^{-(fragmentDistance*fogDensity)^2}$$

În shader-ul de vârfuri, am calculat poziția obiectului și distanța relativă față de camera de vizualizare, iar ulterior acestea au fost integrate în formula menționată anterior.

Densitatea ceții crește în timp până la atingerea unei anumite valori atunci când apăsăm tasta "f". După ce atinge acea valoare, la apăsarea din nou a tastei "f", densitatea începe să scadă în timp.



- **Lumina punctiformă**

Pentru lumina punctiformă, în cadrul fragment shader-ului există funcția "CalcPointLight()". Pentru implementarea acestei tipuri de lumini, am avut nevoie de trei parametri (constant, linear și quadratic), preluați din sursa principală și transmiși funcției din fragment shader, și de un vector care să specifice poziția luminilor. Cu cât acești trei parametri au valori mai mici, cu atât intensitatea luminii crește. Aceste valori pot fi

modificate de la tastatură, oferind impresia unei creșteri a intensității luminii. Factorul constant rămâne la valoarea 1 și sunt modificate doar linear și quadratic, inițializate la 0.35 și 0.44. Atenuarea este calculată astfel:

$$Attenuation = 1 / (Constant + Linear * Distance + Quadratic * Distance^2)$$

iar acest coeficient va fi înmulțit cu fiecare componentă: ambientala difuza și speculară.

- **Mișcarea camerei**

Mișcarea camerei pentru explorarea scenei se realizează folosind mouse-ul și tastatura. Rotația camerei este controlată cu ajutorul mouse-ului, iar deplasările la stânga sau la dreapta, sau înainte și înapoi sunt gestionate cu tastatura. Pentru rotația camerei cu ajutorul mouse-ului, există în funcția principală "mouseCallback", care preia coordonatele inițiale ale mouse-ului și apoi efectuează rotația camerei în funcție de mișcările mouse-ului și coordonatele pointer-ului acestuia.

Pentru navigarea în scenă cu ajutorul tastaturii, a fost implementată în "Camera.cpp" funcția "move", care primește ca și parametri direcția de mișcare a camerei și viteza cu care aceasta se deplasează în direcția indicată de primul parametru. În interiorul acestei funcții, cu ajutorul celor doi parametri, se vor modifica parametrii camerei: "cameraPosition" și "cameraTarget", în funcție de direcția în care se dorește deplasarea camerei.

- **Animații**

Ca animații am realizat mișcarea sus-jos a mingii de ping-pong pe masa de ping-pong, efectul de plutire a unei scânduri prin folosirea mișcărilor sus-jos și interacțiunea în scenă prin posibilitatea persoanei care ține camera să ia paharul de cafea în mână.



3.1.1. Soluții posibile

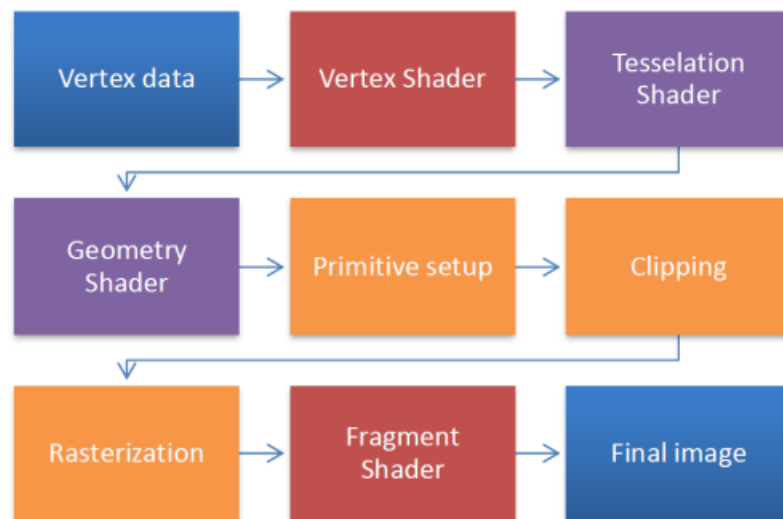
Pentru a implementa iluminatul pe scenă, se pot utiliza diferite tipuri de lumini, cum ar fi lumina direcțională, punctiformă și spotlight. Pentru efecte speciale, se pot include opțiuni precum generarea dinamică de obiecte, detectarea coliziunilor, generarea umbrelor, ceața, ploaia sau vântul.

3.1.2. Motivarea abordării alese

Am ales să implementez pentru scenă: lumini punctiforme, care pot fi activate sau dezactivate prin intermediul tastaturii. Pentru un plus de realism, am inclus algoritmul de generare a ceții, deoarece consider că se potrivește cu aspectul scenei.

3.2. Modelul grafic

Pentru implementare, am utilizat modelul grafic definit de OpenGL. Am programat doar Vertex Shader și Fragment Shader, care se ocupă de calculul iluminării și generarea ceții.



3.3. Structuri de date

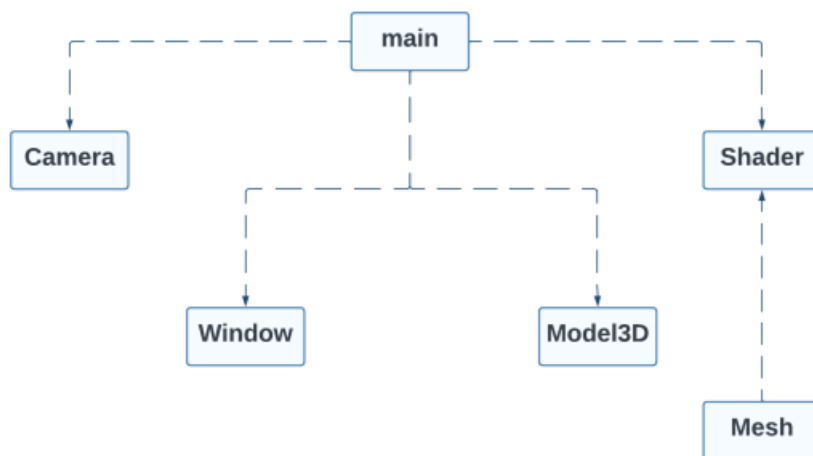
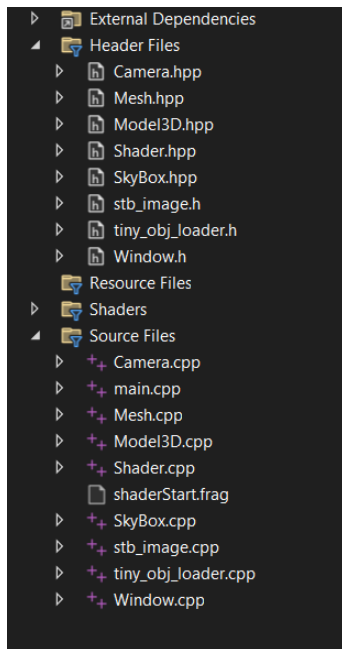
Structurile de date folosite sunt:

- vec<n> și mat<n>: disponibile în biblioteca GLM
- GLint, GLfloat, GLFWwindow, GLenum: disponibile în OpenGL
- Camera, Shader, Model3D, Window: disponibile în proiectul de bază
- PointLightParams: o structură creată pentru a îngloba caracteristicile unei lumini punctiforme

o aceasta include:

- vec3 position
- vec3 color
- GLfloat constant
- GLfloat linear
- GLfloat quadratic

3.4. Ierarhia de clase



4. Prezentarea interfeței grafice utilizator/ manual de utilizare

Pentru a explora scena, utilizatorul are la dispoziție mai multe opțiuni, pe care le poate selecta folosind diverse comenzi. Iată cum pot fi utilizate metodele de navigare:

Navigare cu tastatura:

- Apăsarea tastei "A" va deplasa camera în stânga.
- Apăsarea tastei "D" va deplasa camera în dreapta.
- Apăsarea tastei "W" va deplasa camera înainte (efect de zoom-in).
- Apăsarea tastei "S" va deplasa camera înapoi (efect de zoom-out).

Navigare cu mouse-ul:

- Mișcarea mouse-ului înainte și înapoi va deplasa camera în sus și în jos, schimbând direcția de privire a scenei.
- Mișcarea mouse-ului la stânga și la dreapta va roti camera în jurul axei Y pentru a vizualiza scena în toate direcțiile.

Alte comenzi și funcționalități:

- Apăsarea tastei "1" pentru a vizualiza scena în mod solid.
- Apăsarea tastei "2" pentru a vizualiza scena în mod wireframe.
- Apăsarea tastei "3" pentru a vizualiza scena în mod punctiform.
- Apăsarea tastei "N" pentru a muta în modul noapte.
- Apăsarea tastei "Z" pentru a reveni la modul zi.
- Apăsarea tastei "F" pentru a porni/opri ceața. Ceața poate fi oprită doar când densitatea acesteia atinge un anumit prag pentru a evidenția efectul de ceață.
- Apăsarea tastei "C" pentru a porni animația camerei.

Aceste comenzi oferă utilizatorului control asupra modului în care explorează și interacționează cu scena 3D, permitându-i să ajusteze perspectiva, să schimbe modurile de vizualizare și să exploreze funcționalități specifice ale proiectului.



5. Concluzii și dezvoltări ulterioare

Proiectul actual oferă o gamă variată de efecte, însă există potențial pentru dezvoltări viitoare prin:

Creșterea dimensiunii scenei:

- Extinderea spațiului disponibil în scenă pentru a permite includerea unor peisaje mai ample sau detalii extinse.

Adăugarea de obiecte suplimentare:

- Integrarea unor noi obiecte sau elemente pentru a îmbogăți conținutul vizual și pentru a oferi mai multe opțiuni de explorare.

Implementarea umbrelor:

- Dezvoltarea unui sistem de umbre pentru a adăuga realism și a îmbunătăți percepția adâncimii în scenă.

Adăugarea de noi animații:

- Introducerea de animații suplimentare pentru obiecte, mediu sau chiar pentru camera utilizatorului, pentru a crea o experiență mai dinamică și captivantă.

Detectarea coliziunii:

Implementarea unui mecanism de detecție a coliziunilor pentru a permite interacțiuni mai avansate între obiecte sau între utilizator și mediu.

Aceste posibile extinderi ar contribui la evoluția și îmbunătățirea proiectului, oferind utilizatorilor o experiență mai complexă și captivantă în cadrul scenei 3D. Astfel, proiectul poate fi adaptat pentru a include mai multe elemente, interactivitate și realism, consolidându-și astfel atractivitatea și funcționalitatea în viitor.

6. Referințe

<https://learnopengl.com/Introduction>

<https://free3d.com/>

<https://www.cgtrader.com/free-3d-models>

<https://www.textures.com/library>