



UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

Prezentare fotorealistica a unei scene cu obiecte 3D

Prelucrare Grafica

Autor: Istrate Cristian Andrei

Grupa: 30237

FACULTATEA DE AUTOMATICA
SI CALCULATOARE

2024

Cuprins

1 Prezentarea temei	2
2 Scenariul	2
2.1 Descrierea scenei si a obiectelor	2
2.2 Functionalitati	2
3 Detalii de implementare	4
3.1 Functii și Algoritmi	4
3.1.1 Solutii posibile	4
3.1.2 Motivarea abordarii alese	6
3.2 Modelul grafic	6
3.3 Structuri de date	6
3.4 Ierarhia de clase	6
4 Prezentarea interfetei grafice utilizator / manual de utilizare	6
5 Concluzii si dezvoltari ulterioare	7
6 Referinte	7

1 Prezentarea temei

In aceasta aplicatie avem de realizat implementarea unei scene din obiecte 3D cu ajutorul unui program de modelare si anume Blender in care cream obiectele pe care dorim sa le incarcam in scena noastra. Scena este incarcata in OpenGL care este un standard pentru programarea componentelor grafice 2D si 3D. Codul de OpenGL este in limbachul C++ si cea mai mare parte din el a fost realizata in Microsoft Visual Studio. Pentru realizarea scenei, suntem la libera creatie, iar pe urma avem de implementat diversi algoritmi pentru imbunatatirea acestieia.

2 Scenariul

2.1 Descrierea scenei si a obiectelor

Ca scena am ales realizarea unui cartier de case dintr-o vale. Acest cartier se afla la o intersecție de drumuri pe care am adaugat diverse masini cu diverse texturi, scena fiind predominata de vehicule precum si de un elicopter.



Figura 1: Scena

Obiectele sunt toate texturate si diversificate, scena dispune de un numar mare de texturi precum 8 case diferite, 5 tipuri de masini cu texturi diversificate si realizeate custom, existenta unei parcuri, a unui lac, a unui teren de fotbal precum si o mica lucrare unde se afla un excavator care sapa. Avem de asemenea doua tipuri de copaci aranjati pe harta, terenul de fotbal contine o minge, doua porti, dungile trase pe jos. Pe langa drumuri avem trotuare, cosuri de gunoi, felinare stradale. Langa lac avem de asemenea niste bancute care ofera un aspect aparte.

2.2 Functionalitati

In scena este posibila deplasarea cu ajutorul tastelor W A S D precum si a mouse-ului pentru a merge in diverse locuri ale scenei, iar in ceea ce priveste functionalitatile mai avansate avem:

- Schimbarea lumинii pentru trecerea de la zi la noapte cu ajutorul tastelor K si L.



Figura 2: Scena pe noapte

- Adaugarea sau reducerea cetii cu ajutorul tastelor M si N.



Figura 3: Scena in ceata

- Rotatia unghiului din care vine lumina pentru a modifica directia umbrelor cu ajutorul tastelor Z si X.
- Pornirea sau oprirea ploii din scena cu ajutorul tastei R.



Figura 4: Scena in ploaie

- Trecea in modurile Solid / Wireframe si Poligonal prin tasta P.

3 Detalii de implementare

3.1 Functii și Algoritmi

3.1.1 Solutii posibile

Pentru realizarea acestei aplicatii am implementat diversi algoritmi care imbunatatesc gradul de fotorealism al aplicatiei precum si nivelul de detaliere al scenei.

Din punctul de vedere al surselor de lumina scena noastră dispune de două surse de lumina principale care au fost implementate în Shaderul 'basic'. Lumina noastră principală este una directională care în scenă de fata provine de la soare și a fost adaptată în cod pentru a putea modifica intensitatea acesteia. A doua sursă de lumina de care dispune este lumina PointLight care a fost setată la fiecare lumina stradală din scenă și se aprind automat în momentul în care scenă trece de un anumit prag de luminositate facându-se noapte.

De asemenea scenă dispune de umbre care sunt influențate de lumina directională, iar acestea sporesc cu mult nivelul de fotorealism, iar fiecare obiect din scenă dispune de propria umbra indiferent dacă obiectul este unul dinamic sau nu.

Pe lângă cele două modificări aduse shader-ului și punerea lor în funcțiune în cod, scenă noastră dispune și de 4 animații dintre care:

1. Animatia unei mașini care execută un derapaj controlat în giratoriul din scenă.

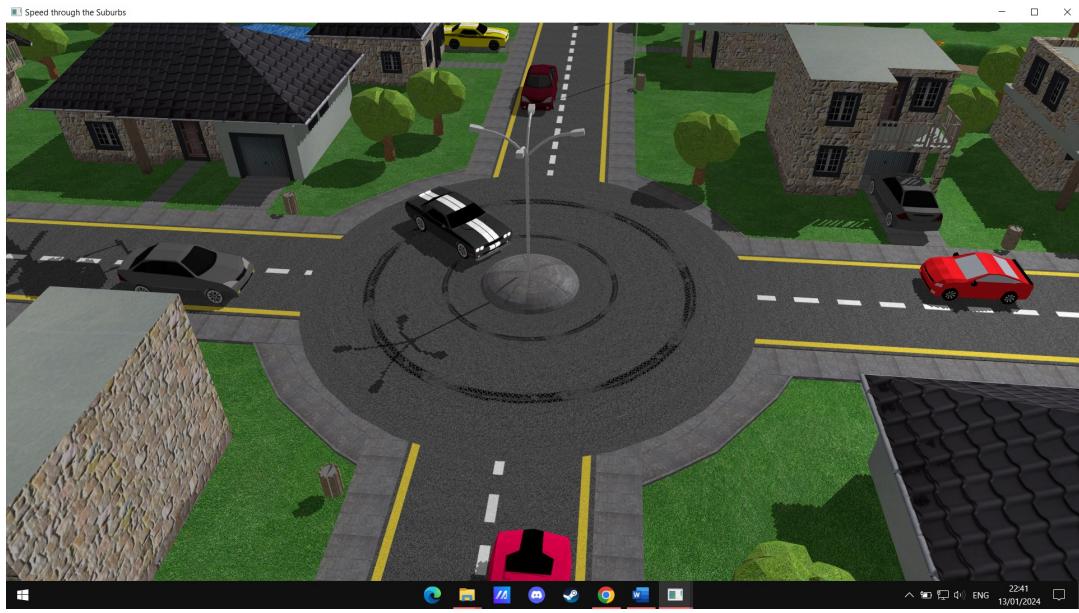


Figura 5: Animatie derapaj

2. Elicea cea mare a elicopterului care se afla pe langa giratoriu.
3. Elicea cea micuta a elicopterului care se afla pe o axa diferita de cea mare si urmarest tot o rotatie.

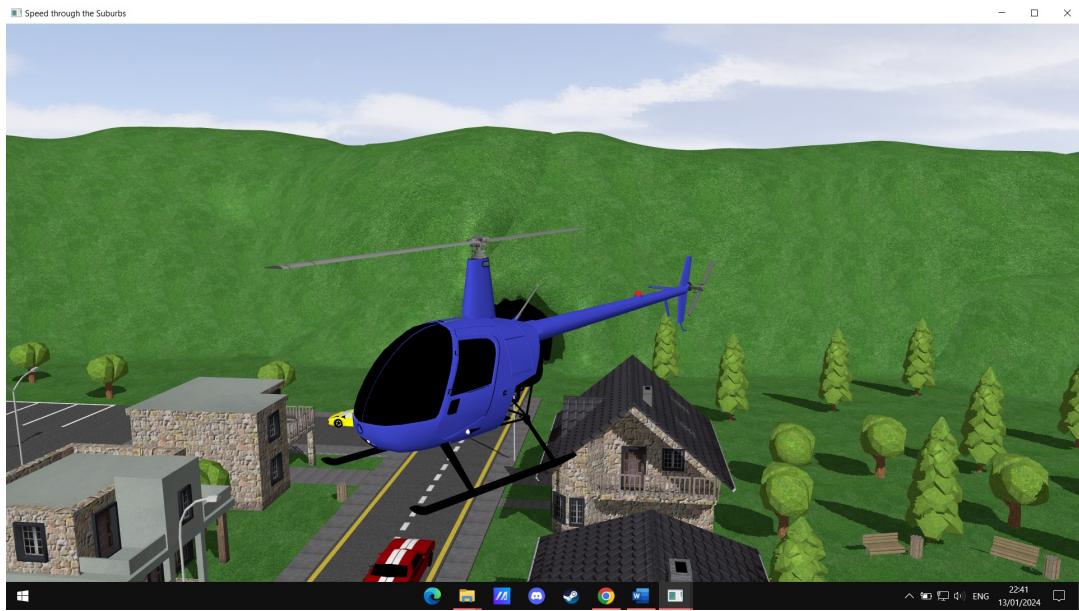


Figura 6: Animatie elicopter

4. Animatia de inceput a camerei care porneste odata cu aplicatia si se apropie de centrul hartii de unde porneste spre cealalta parte si se ridica spre cer adaugand o imagine de ansamblu a scenei.

Efectul de ploaie prezent in scena de asemenea activabil pe una dintre taste aduce o nota aparte scenei si o functionalitate in plus, este realizata cu ajutorul unui obiect drept picatura de ploaie care este generat prin scena la diferite coordonate aducand efectul de ploaie.

Tot pentru fotorealism am adaugat si efectul de ceata care poate fi modificat de utilizator.

3.1.2 Motivarea abordarii alese

Am ales aceste implementari pentru a imbunatatii vizual scena realizata si pentru a-i sporii gradul de realism. Algoritmii implementati au urmarit modelul laboratoarelor, iar majoritatea implementarilor au fost realizate cu informatiile si resursele oferite.

3.2 Modelul grafic

Aplicatia este constituita dintr-un obiect mare care contine marea majoritatea a obiectelor din scena mai putin cele animate. Obiectele care dispun de animatii au fost incarcate separat si translatate, rotite si aranjate in functie de scopul lor. Avem 3 shadere pe care le folosim, cel mare basic in care sunt calculate lumiile, umbrele si ceata, skyboxShader cu ajutorul caruia desenam cele doua skybox-uri prezente in scena noastra care se regasesc si ele in folderul radacina si deptMapShader care ne ajuta in momentul calcularii hartii de adancime atunci cand desenam umbrele. Fiecare obiect din scena dispune de o imagine care constituie textura acestuia si se regaseste tot in folderul radacina. Avem obiecte care emit lumina si toate obiectele din scena dispun de propria umbra.

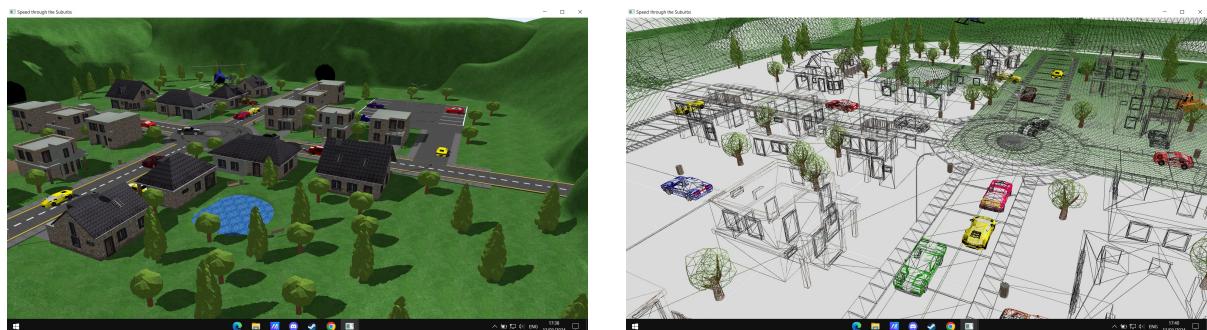
3.3 Structuri de date

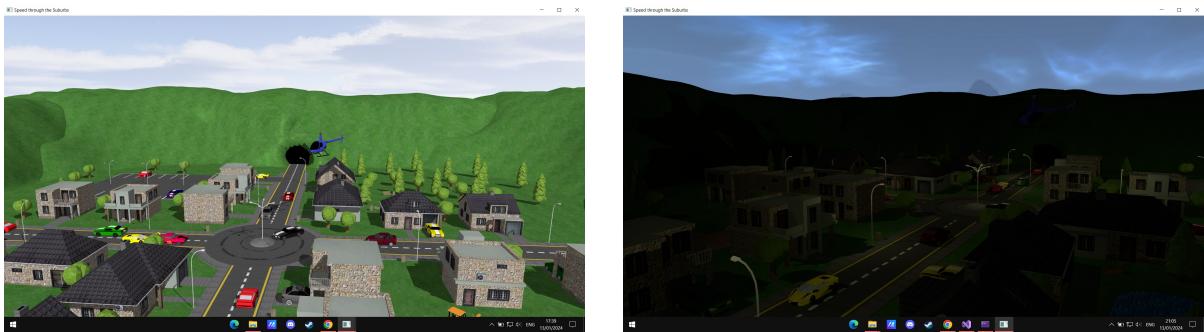
Ca structuri de date s-au folosit librariile oferite de OpenGL precum GLFW, GLM si GL. Care dispun de diverse tipuri precum mat4, mat3, vec3 etc si functiile aferente acestor tipuri de date care ajuta la realizarea diverselor operatii de care avem nevoie in aplicatie.

3.4 Ierarhia de clase

- main - in care se realizeaza in principal aplicatia noastra.
- camera - care se ocupa de miscarile camerei.
- shader - pentru incarcarea diverselor shadere de care avem nevoie si lucrul cu ele.
- skybox - pentru incarcarea cerului din scena
- Diverse clase necesarea pentru buna functionare a aplicatiei nostre OpenGL precum window, stb_image, etc.

4 Prezentarea interfetei grafice utilizator / manual de utilizare





Pentru a porni aplicatia trebuie accesat executabilul "ProiectOpenGL.exe" din folderul radacina si astfel ne putem deplasa prin scena si folosii functionalitatile de care dispunem.

5 Concluzii si dezvoltari ulterioare

In concluzie aplicatia ne ajuta pentru intelegera mediului OpenGL si adaptarea cu acesta. In procesul dezvoltarii acestei aplicatii am reusit sa intelegh diverse aspecte de modelarea 3D, de lucrul cu obiectele precum si functionarea shaderelor si posibilele dezvoltari a acestora.

In viitor am putea implementa diverse animatii a masinilor care se pot deplasa prin scena, adaugarea mult mai multor obiecte pentru complexitatea scenei. Am putea adauga si diverse efecte pe langa ceata deja existenta, cum ar fi efectul de ploaie, ninsoare etc. Iar una din cele mai mari imbunatariri aduse ar putea fi detectia coliziunilor cu obiectele din scena.

6 Referinte

- <https://www.free3d.com/>
- <https://www.blendswap.com/>
- <https://www.turbosquid.com/>
- <https://ambientcg.com/>
- <https://learnopengl.com/>
- <https://www.humus.name/index.php?page=Textures&start=0>