

Documentatie Project- Proiectare Grafica



Cuprins

1. Specificarea subiectului.....	2
2. Scenariu.....	3
2.1 Scena si descrierea obiectelor.....	3
2.2 Functionalitati.....	3
3. Implementarea detaliilor.....	7
3.1 Functii si algoritmi.....	7
3.1.1 Solutii posibile.....	7
3.1.2 Motivația alegerii acestei abordări.....	9
3.2 Modelul grafic.....	9
3.3 Structura datelor.....	9
3.4 Ierarhia Claselor.....	9
4. Modul de folosire al interfetei.....	9
5. Concluzii si dezvoltari viitoare.....	10
6. Referinte.....	10

1. Specificarea Subiectului

Acet proiect demonstrează crearea unei scene interactive 3D a unui spatiu verde, dedicat relaxarii, în OpenGL, evidențiind integrarea unor funcționalități grafice avansate și modele detaliate realizate în Blender, într-un mediu explorabil în timp real. Prin încorporarea efectelor meteorologice, iluminării, umbrelor și animațiilor de obiecte, proiectul își propune să atingă un nivel ridicat de fidelitate vizuală și implicare.

2. Scenariu

2.1 Scena si descrierea obiectelor

Peisajul proiectului prezintă un spațiu verde simplu și plăcut, care include o căsuță/cabana mica cu o canapea, un TV, o masa din sticlă și cîteva lampi ce dău o nota de ambient scenei, un lac pe care am pus niste ratuste, mulți brazi printre care am pus felinare, flori și cîteva elemente animate, cum ar fi câinele care se învârte în jurul casei. Toate obiectele au fost create în Blender și sunt gândite să pară cât mai realiste, de la detaliile lacului până la texturile copacilor și ale căsuței. Am acordat atenție modului în care arată materialele și texturile pentru ca scena să fie cât mai naturală și plăcută vizual.

2.2 Functionalități

Lumina de 2 tipuri :directionala și punctiformă

Efect special: ceata

Animarea Obiectelor: un caine ce se învârte în jurul casei

Animarea camerei: navigarea controlată de utilizator, dar și o animație de prezentare a scenei

Diferite modele de rendering: solid, wireframe, polygonal și smooth

Modul de noapte: simulează noaptea peste scenă mea

Exemplificarea tipurilor de lumini

Directionala:



Punctiforma:



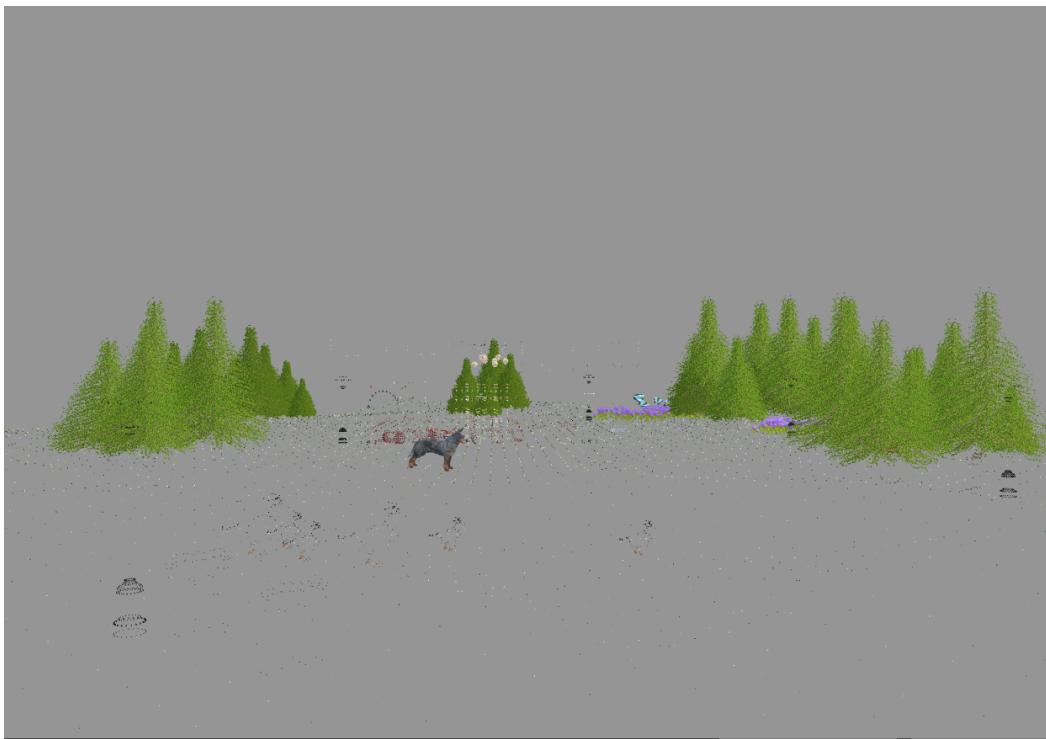
Efectul de ceata



WireFrame



Poligonal



Smooth



3. Detalii de implementare

3.1 Functii si algoritmi

3.1.1 Solutii posibile

Sursele de lumină:

1. Lumina direcțională (lightDir):
 - Această lumină simulează o sursă de lumină mare, cum ar fi soarele.
 - Lumina este uniformă și vine dintr-o anumită direcție specificată de vectorul lightDir.
 - Este folosită pentru a ilumina întreaga scenă într-un mod consistent, adăugând atât iluminare ambientală (lumina generală), cât și lumina difuză și speculară (pentru efecte de luciu).

2. Luminile de tip punct (lightPos1, lightPos2, etc):
 - Aceste lumini simulează surse de lumină mici și localizate, cum ar fi lămpi, felinare sau alte obiecte luminoase.
 - Poziția fiecărei surse de lumină este specificată prin coordonatele sale (ex. lightPos1, lightPos2 etc.).
 - Luminile de tip punct au un efect care scade odată cu distanța față de sursă, datorită unui factor de atenuare calculat folosind constantele constant, linear și quadratic. Acest lucru face ca lumina să fie mai realistă.

Utilizări în scenă:

- lightPos1 și lightPos7: Reprezintă lămpi .
 - lightPos2 - lightPos6: Reprezintă felinarele de-a lungul scenei, care adaugă un plus de detaliu și atmosferă, mai ales în condiții de lumină redusă sau pe timp de noapte.
-
3. Lumina de noapte (night):
 - Când este activată (night == 1), toate sursele de lumină din scenă sunt închise, iar fragmentul este redat complet întunecat (vec4(0.0, 0.0, 0.0, 1.0)).
 - Acest mod poate fi util pentru a simula tranziția între zi și noapte.

Ceață:

- Ceață este implementată pentru a adăuga profunzime scenei și pentru a simula condiții atmosferice, cum ar fi un peisaj cețos sau distanțe mari unde obiectele devin mai puțin clare.
- Este controlată de uniforma fogDensity, care determină cât de densă este ceață.
- Fragmentele mai îndepărivate de cameră sunt afectate mai mult de ceață, ceea ce creează o tranziție graduală între obiectele din apropiere (clar vizibile) și cele îndepărivate (mai estompate).

1. Cum funcționează:

- Se calculează distanța fragmentului față de cameră (fragmentDistance = length(fPosEye)).
- În funcție de această distanță și de densitatea cești, se determină un factor de ceată (fogFactor), care controlează cât de mult influențează ceata culoarea finală.
- Culoarea cești este definită prin variabila fogColor (un gri neutru în acest caz: vec4(0.5, 0.5, 0.5, 1.0)).

Utilizare în scenă:

- Ceata este folosită pentru a crea o atmosferă mai realistă, făcând obiectele aflate la distanță să pară mai estompate. Acest efect îmbunătăște imersiunea vizuală, în special într-o scenă naturală cu lac, copaci și felinare.

Biblioteca OpenGL include o gamă largă de funcții, printre care:

glViewport(...): folosită pentru configurarea transformării viewport-ului.

glfwCreateWindow(...): utilizată pentru crearea ferestrelor.

glGenTextures(...), glBindTexture(...), glTexImage2D(...): funcții pentru generarea și manipularea texturilor, inclusiv crearea texturilor de adâncime pentru obiectele de tip Framebuffer. Si multe altele

În cadrul acestui proiect, funcția principală **void renderScene()** gestionează trimiterea datelor către shadere. Aceasta include crearea modelelor și umbrelor, calculul matricii normale, precum și operațiile de rotație, translație și scalare.

De asemenea, funcția **void initUniforms()** este esențială pentru configurarea proprietăților luminii, incluzând lumina punctiformă, direcțională și spot. Funcția **void initShaders()** initializează programele shader, iar **void initModels()** creează toate modelele 3D necesare pentru scenă.

Alte funcții importante sunt:

void processMovement(): gestionează mișcarea obiectelor.

void move(gps::MOVE_DIRECTION direction, float speed): controlează direcția și viteza mișcării.

void mouseCallback(GLFWwindow window, double xpos, double ypos): gestionează mișcarea mouse-ului.

void keyboardCallback(GLFWwindow window, int key, int scancode, int action, int mode): gestionează interacțiunile de la tastatură.

void scroll_callback(GLFWwindow window, double xoffset, double yoffset): această funcție gestionează acțiunile derulării (scroll) pentru a ajusta, de exemplu, zoom-ul camerei sau alte setări care necesită un parametru ajustabil cu ajutorul derulării mouse-ului.

Aceste funcții permit utilizatorului să controleze mișcarea modelelor sau a camerei. Funcția **mouseCallback()** folosește unghiuri Euler pentru a interpreta mișările mouse-ului, iar operațiile matematice și algoritmii necesari sunt realizate cu ajutorul bibliotecii GLM (OpenGL Mathematics).

3.1.2 Motivația alegerii acestei abordări

În cadrul laboratoarelor dedicate acestui curs, am învățat să calculăm lumina, umbrele și să aplicăm operații de translație, scalare și rotație pe diverse obiecte. Cu această bază teoretică și având deja un schelet de proiect, sarcinile mele au fost clar definite, oferindu-mi o direcție bine structurată pentru finalizarea proiectului.

3.2 Modelul Grafic

Procesul de căutare a obiectelor și texturilor de pe internet a fost destul de dificil, mai ales în situațiile în care texturile nu se afișau corect pe obiecte sau când obiectele nu apăreau aşa cum mă așteptam. Am petrecut mult timp căutând obiecte și texturi potrivite. Pentru unele dintre acestea, am realizat personal texturarea. În ciuda acestor provocări, am considerat această parte a proiectului ca fiind creativă și plăcută.

3.3 Structuri de Date

Structurile de date necesare au fost destul de simple, fiind deja prezентate și implementate în cadrul lucrărilor de laborator. Astfel, munca mea a fost semnificativ simplificată. Totuși, a fost nevoie să creez câteva funcții suplimentare pentru a calcula necesitățile pentru modul de noapte, ceață, lumini și alte efecte.

3.4 Ierarhia Claselor

Proiectul are o abordare orientată pe obiecte, utilizând clase pentru diverse funcționalități precum controlul camerei, animații, iluminare, randare, skybox, Model3D, Shader și altele.

4. Modul de folosire al interfetei

Utilizatorul se poate folosi de următoarele taste pentru navigație în scenă

- W - mută camera în față
- S - mută camera în spate
- A - mută camera la stânga
- D - mută camera la dreapta
- U - mișcă camera în sus
- I - mișcă camera în jos
- Q - rotirea camerei spre stânga
- E - rotirea camerei spre dreapta
- N - activează și dezactivează night mode
- F - activează ceata, cu cat apesi mai lung pe tasta cu atât crește intensitatea cetii
- G - dezactivează ceata, cu cat apesi mai lung pe tasta cu atât scade intensitatea cetii
- L - activează luminile punctiforme, adică lampile și felinarele
- P - activează wireFrame-ul
- O - activează poligonal view

- K - activeaza solid view
- L - activeaza smooth view
- B - activeaza animatia de prezentare

5. Concluzii

In concluzie aş dori să spun că am considerat destul de dificil să lucrez cu Blender, mai ales pentru că a fost prima dată când am folosit acest software. Laptopul meu a întâmpinat câteva probleme în timpul utilizării, cum ar fi blocări neașteptate. Cu toate acestea, am descoperit că lucrul cu Blender poate fi foarte util și satisfăcător, odată ce te obișnuiești cu el.

În ceea ce privește partea de programare, am găsit-o destul de complicată. Am petrecut mult timp încercând să înțeleg structurile de cod și să lucrez cu ele. De asemenea, procesul de depanare manuală a fost dificil, dar absolut necesar.

Acest proiect a necesitat un efort și un timp considerabil, dar finalizarea lui a fost un success și am fost mandra ca am reusit să il aduc într-un punct în care poate fi predat.

Dezvoltări ulterioare:

-ploaia și vantul

-umbrirea propriu zisă a obiectelor

-adăugarea mai multor animații

-adăugarea mai multor obiecte, pentru a rezulta o scenă mai complexă

6. Referințe

- Lucrarile de laborator și cursurile
- <https://learnopengl.com/>
- <https://www.turbosquid.com/>
- <https://www.cgtrader.com/>