



Documentatie Proiect

Sisteme de Prelucrare Grafica

Halasz Lorand Daniel
Grupa 30236

Cuprins

1. Prezentarea temei.....	3
2. Scenariul.....	4
2.1. Descrierea scenei si a obiectelor	4
2.2. Functionalitati	7
3. Detalii de implementare	8
3.1. Functii si algoritmi.....	8
3.1.1. Solutii posibile	8
3.1.2. Motivarea abordarii alese	9
3.2. Structuri de date	9
3.3. Ierarhia de clase.....	10
4. Prezentarea interfetei grafice utilizator / manualul de utilizare	11
5. Concluzii si dezvoltari ulterioare.....	15
6. Referinte	16

1. Prezentarea temei

Proiectul are ca si scop realizarea unei prezentari fotorealiste a unor scene de obiecte 3D utilizand librariile prezentate la laborator (OpenGL, GLFW, GLM, ...). Proiectul trebuie sa respecte urmatoarele aspecte:

- Posibilitatea de a vizualiza scena prin miscarea camerei, utilizand tastatura, mouse-ul si animatii de prezentare
- Specificarea a cel putin doua surse de lumina
- Vizualizarea scenei in diferite moduri
- Maparea unor texturi cat mai detaliate
- Exemplificarea generarii umbrelor
- Exemplificarea animarii diferitelor componente ale obiectelor
- Crearea unei scene cat mai fotorealiste, complexe si cu un nivel ridicat de detaliere
- Dezvoltarea diferitilor algoritmi si implementarea acestora

2. Scenariul

2.1. Descrierea scenei si a obiectelor

Scena proiectului este realizata cu ajutorul aplicatiei Blender, in cadrul careia s-au mapat texturile pe obiecte si, de asemenea, s-a construit ground-ul scenei, care este compus pe de o parte de un plan care este acoperit de nisip, iar pe de alta parte, de o zona cu apa.

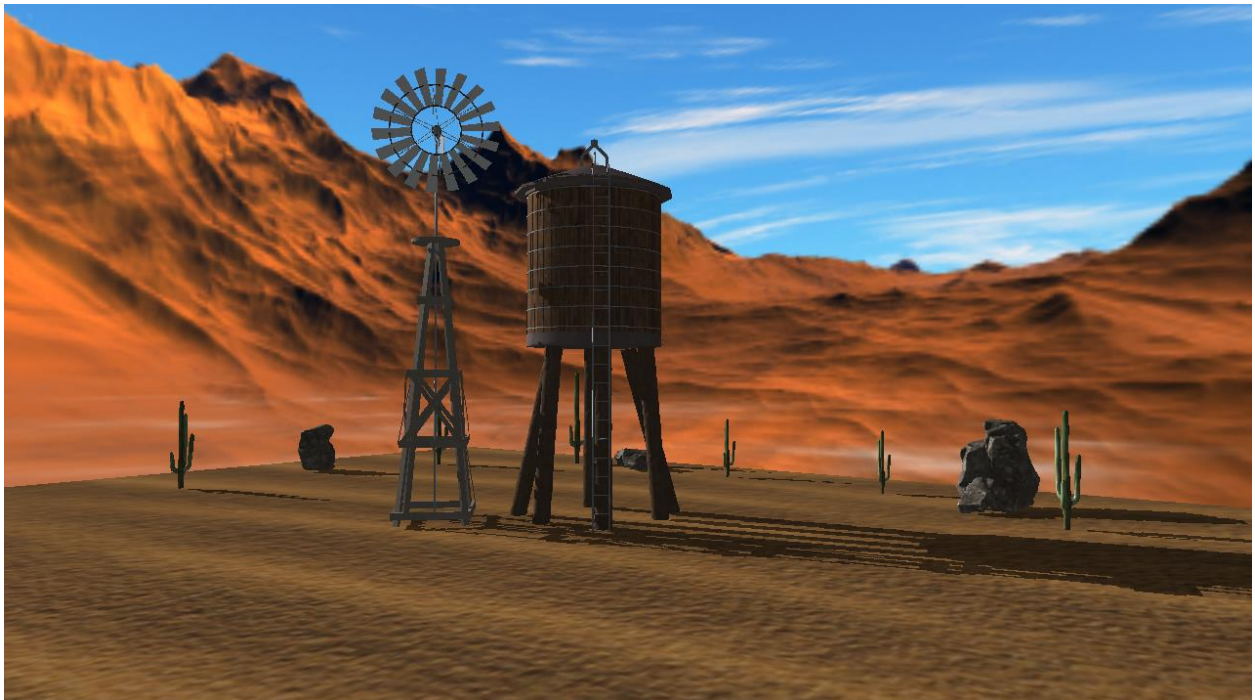
Scena este compusa din diferite obiecte distincte, o parte din ele, asezate direct pe plan utilizand aplicatia Blender si formand astfel ground-ul scenei, iar o alta parte, asezate in scena cu ajutorul bibliotecii OpenGL, care impreuna compun o scena desprinsa din desert.

Obiectele, care sunt asezate in scena direct in Blender, sunt acele obiecte statice, care compun decorul acesteia si care nu au nicio functionalitate in scenariul proiectului. In schimb, obiectele care indeplinesc un rol in acest scenariu sunt incarcate separat.

In cadrul scenei avem urmatoarele obiecte:

- Ground – reprezinta scena propriu-zisa, care este defapt o zona de desert in cadrul careia se afla o oaza. Aceasta este realizata in Blender si contine urmatoarele obiecte, in afara celor doua plane (nisip si apa):
 - Piramida
 - Sfinx
 - Vegetatie (diferite tipuri de palmieri si cactusi)
 - Stanci
 - Animale (canguri si camile)
- Jeep, tireFL, tireFR, tireBL, tireBR – toate la un loc compun o masina de teren
- WindMill cu componentele ei (weathervane si ventilator) – care reprezinta o moara de vant si un rezervor de apa

Prezentarea scenei:







2.2. Functionalitati

In cadrul proiectului, utilizatorul are urmatoarele posibilitati:

- de a vizualiza scena fotorealista, cu ajutorul miscarii camerei, avand 2 moduri
- de a vizualiza scena utilizand o animatie de prezentare
- de a vizualiza scena in modul wireframe sau point
- de a folosi efectele sonore puse la dispozitie
- de a misca sursa de lumina
- de a conduce o masina
- de a creste sau scadea puterea unei furtuni de nisip
- de a vizualiza diferite animatii ale unor obiecte, precum invartirea unei mori de vant in functie de puterea furtunii sau a simularii realiste a miscarii masinii

3. Detalii de implementare

3.1. Functii si algoritmi

3.1.1. Solutii posibile

Pentru iluminarea scenei am folosit modelul Blinn-Phong, care este o versiune usor modificata a modelului Phong si a fost dezvoltat de catre Jim Blinn. Scopul sau este de a imbunatati reflexiile speculare in anumite conditii. Solutia propusa de Blinn este de a folosi semi-vectorul (half-vector), in locul vectorului de reflexie pentru a calcula componenta speculara. Semi vectorul este un vector unitar aflat la jumatatea distantei dintre directia de vizionare si directia luminii. In cadrul proiectului exista 2 surse de lumina, una de tip directional si una de tip punctiform.

Umbrele adauga un grad de realism scenei, permitand, de asemenea, o perceptie mai profunda a adancimii. Pentru crearea umbrelor am folosit tehnica Shadow Mapping, implementata cu metoda depth-bufferului. Aceasta este o tehnica care utilizeaza texturi de adancime pentru a decide daca un punct se afla sau nu in umbra. Cheia acestui algoritm, consta in crearea unei harti de adancime din perspectiva sursei de lumina, dupa care se rasterizeaza scena din punctul de vedere al observatorului, comparandu-se adancimea fiecarui fragment vizibil cu valorile de adancime din harta umbrelor. Fragmentele care au o adancime mai mare decat cea care a fost stocata anterior in harta de adancime nu sunt direct vizibile din punctul de vedere al luminii si sunt, prin urmare, in umbra.

Pentru cresterea realismului in scena 3D, am adaugat un efect care are scopul de a simula o furtuna de nisip si de a crea o atmosfera speciala. Prin manipularea atributelor asociate acestuia putem sa personalizam atmosfera si, de asemenea, sa amelioram perceptia de adancime.

De asemenea, tot pentru cresterea realismului in scena, am ales adaugarea unui cubemap (denumit skybox), pentru a afisa un background pe toate zonele din scena 3D pe care nu le-am modelat. Acesta cub este alcatuit din 6 texturi, care mapate in mod corespunzator creeaza senzatia realitatii.

Miscarea in scena este creata cu ajutorul clasei Camera prin intermediul unor metode, precum "move", "rotate" sau "setCameraPosition". In cadrul aplicatiei exista doua moduri de camera, una asemanatoare cu camera oricarui joc, care nu are posibilitatea de a depasi granitele scenei, si nici de a intra in coliziune cu alte obiecte, iar la cel de-al doilea mod avand posibilitatea de a ne deplasa cu camera peste tot. Primul mod a fost realizat prin limitarea deplasarii camerei pe axa z, respectiv prin utilizarea unei functii care detecteaza coliziunile prin compararea coordonatelor camerei cu a celorlalte obiecte si a marginilor scenei.

Animatiile au fost create prin utilizarea unor simple operatii de translatii si rotatii. In cazul miscarii masini, pozitia rotilor este calculata in functie de pozitia masinii, iar pentru miscarea propriu-zisa s-a folosit operatia de translatie, noua pozitie fiind calculata cu ajutorul functiilor trigonometrice sinus si cosinus.

3.1.2. Motivarea abordarii alese

In cazul modelului de iluminare, am ales folosirea modelului Blinn-Phong, intrucat acesta imbunatateste reflexiile speculare in anumite conditii, cum ar fi coeficientul foarte scazut de stralucire, rezultand astfel o zona speculara destul de mare. In cazul modelului Phong, reflexiile speculare tind sa fie intrerupte imediat ce unghiul dintre directia de vizualizare si reflexia luminii creste peste 90 de grade. De asemenea modelul Blinn-Phong imbunatateste performanta, evitand calculul costisitor al vectorului de reflexie.

Exista o multime de tehnici moderne care pot fi folosite pentru a produce efectul de umbre. Cu toate acestea, nu exista algoritmi perfecti in timp real pentru acest lucru, fiecare avand avantaje si dezavantaje. Am ales solutia implementarii tehnicii de shadow mapping, intrucat aceasta este o tehnica ce ofera rezultate bune si iese in evidenta prin modul de implementare mai usor decat in cazul altor tehnici.

3.2. Structuri de date

In cadrul proiectului s-au folosit atat tipuri de date specifice limbajului C, cum ar fi bool, int, float si double, cat si tipuri specifice OpenGL, precum GLFWWindow, GLuint, GLfloat, GLenum, vector (de GLchar, pentru stocarea fetelor skybox-ului), vec3, vec4, mat3, mat4, Camera (folosit pentru a stoca datele necesare camerei, precum pozitia acesteia, directia de vizualizare, etc.), Model3D (folosit pentru stoca obiectele care formeaza scena proiectului), Shader, SkyBox si un tip din biblioteca irrKlang ISoundEngine (care a fost utilizata pentru introducerea unor efecte sonore), si anume.

3.3. Ierarhia de clase

Proiectul este divizat in mai multe clase, astfel:

- Camera – clasa folosita pentru a stoca toate datele si metodele necesare camerei
- Model3D – clasa folosita pentru a stoca obiectele care formeaza scena proiectului, precum si metoda Draw utilizata pentru a desena obiectele
- Shader – clasa folosita pentru a incarca si utiliza shadere-le necesare
- SkyBox – clasa cu ajutorul careia se incarca si se deseneaza skybox-ul
- Clasa de baza care contine functia main, cea care lanseaza intreg programul

4. Prezentarea interfeței grafice utilizator / manualul de utilizare

Pentru început este important de știut faptul că deplasarea în scenă se face utilizând tastele W, A, S, D, pentru direcțiile înainte, stanga, înapoi, respectiv dreapta și mouse-ul pentru rotirea camerei.

În cadrul aplicației, există două tipuri de cameră în modul vizualizare, care se selectează utilizând tasta C:

- Primul mod și considerat de bază în aplicația de față și este asemănător cu mișcarea camerei în orice joc, și anume este posibilă mișcarea în orice direcție (înainte, înapoi, stanga, dreapta), având restricția de a nu trece prin obiecte și de a nu părăsi scena.
- Cel de-al doilea mod este util pentru o poziționare mai bună a camerei, dar nu este foarte realist. În acest mod, mișcarea se face asemănător, cu mențiunea că aici camera se poate deplasa și în sus, respectiv în jos. Acest lucru se realizează utilizând tastele, pentru mișcarea propriu-zisă și mouse-ul pentru selectarea direcției.

Pentru a schimba modul de vizualizare în moduri precum wireframe sau point se utilizează tasta B.

Pentru a activa/dezactiva sunetul se folosește tasta M.

În cadrul aplicației, scena poate fi vizualizată, de asemenea, și utilizând o animație de prezentare care poate fi pornită/oprită cu ajutorul tastei P.

Pentru a opri/porni sursa principală de lumină se folosește tasta N.

În cazul în care nu se doresc ca obiectele să lase umbre pe planul scenei, se poate apăsa tasta G.

O altă animație care poate fi utilizată în scenă este data de rotația giruetei, în ambele sensuri, utilizând tastele Q și E.

Pentru a modifica poziția sursei de lumină se utilizează tastele J și L.

În cazul în care se dorește apariția/dispariția unei furtuni de nisip în/din scenă, se utilizează tasta F împreună cu una din tastele + (crește puterea furtunii) și – (scade puterea furtunii).

Pentru a intra în modul de conducere a mașinii se apasă tasta ENTER pentru a intra în aceasta, dar e de menționat faptul că poziția cameră trebuie să fie lângă mașina pentru a o putea conduce. În cazul în care se dorește să se iasă din mașină se apasă din nou tasta ENTER.

Utilizând săgețile, putem deplasa mașina în orice direcție.

Utilizarea tastei SHIFT, mareste viteza de deplasare a masinii, iar utilizarea tastei SPACE pentru a utiliza frana de mana.

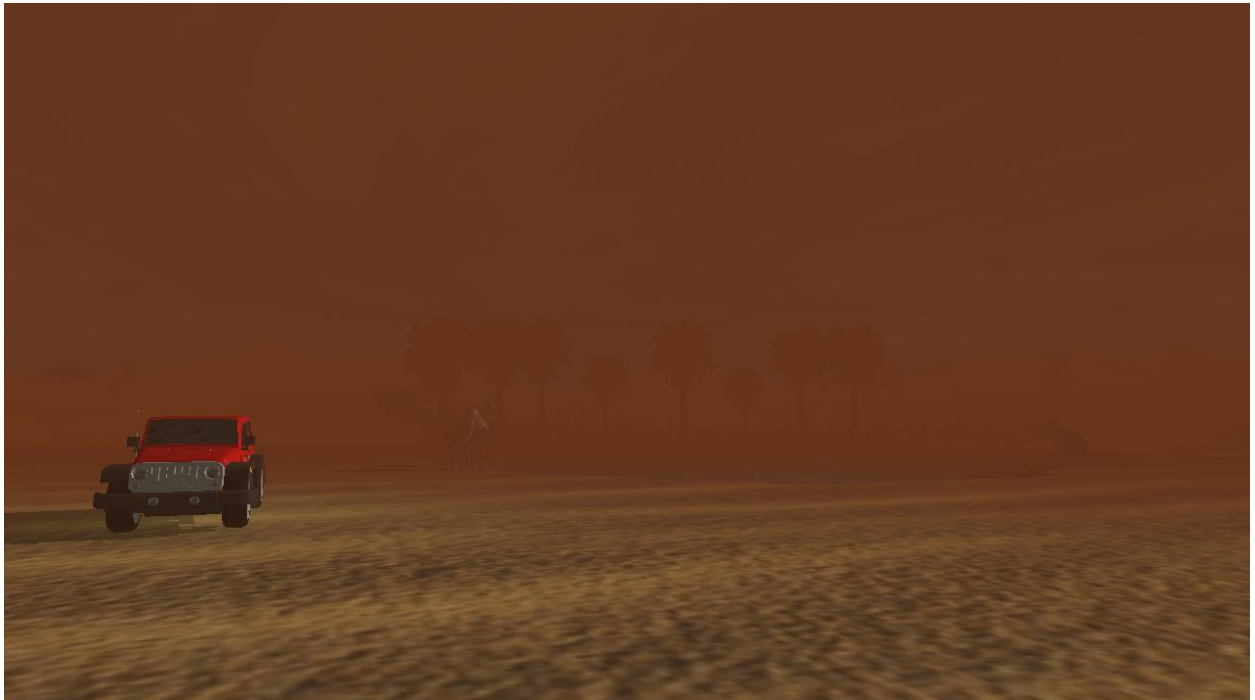
Tot in modul de conducere a masinii, se poate folosi tasta H pentru a claxona, respectiv tasta C pentru a iesi din modul de urmarire a vehiculului si pentru a putea misca separat camera fata de masina.

Redimensionarea ferestrei se poate face utilizand fie mouse-ul, fie tasta F11, pentru a trece din modul window (1280x720) in modul Fullscreen(1920x1080) si viceversa.

Pentru a inchide aplicatia se foloseste tasta ESCAPE.

Legenda:

- Pentru modul de vizualizare:
 - W, A, S, D – deplasare camerei
 - Mouse – miscarea camerei
 - ENTER – intra in masina
 - C – mod camera
 - B – mod vizualizare (wireframe, point)
 - M – mute
 - P – animatie de prezentare
 - N – oprire/pornire sursa de lumina
 - G – mod cu sau fara umbre pe plan
 - Q/E – rotire girueta
 - F + '+' – creste puterea furtunii
 - F + '-' – scade puterea furtunii
 - J/L – miscarea sursei de lumina
 - F11 – Window/Fullscreen
 - ESCAPE – inchiderea aplicatiei
- Pentru modul de conducere:
 - Sageti – miscarea masinii
 - ENTER – iese din masina
 - H – claxon
 - SHIFT – marire viteza
 - SPACE – frana de mana
 - C – mod camera





5. Concluzii si dezvoltari ulterioare

Aplicatia poate fi dezvoltata ulterior intr-o multime de feluri, intrucat aceasta poate reprezenta o baza pentru un proiect mult mai complex. Dintre aceste dezvoltari ulterioare as mentiona urmatoarele:

- Marirea scenei
- Adaugarea unor noi obiecte, pentru a mari nivelul de realism al scenei
- Realizarea unor noi animatii, mult mai complexe, precum miscarea animalelor
- Introducerea unor noi tipuri de lumina
- Introducerea unui caracter, care sa se poata misca
- Implementarea unui efect de vant care sa isi schimbe directia si viteza, astfel incat sa poata determina directia si viteza de rotatie a morii de vant
- Dezvoltarea unor noi functionalitati, precum posibilitatea de a intra in piramida
- Adaugarea mai multor efecte sonore
- Crearea unor noi animatii de prezentare
- Eficientizarea codului

6. Referinte

<https://learnopengl.com/>

<https://moodle.cs.utcluj.ro/>

https://docs.google.com/document/d/1njtWPMmOQNlaD_z9ve8iPRUqQTwdIV_PO-NvPD0nOuM/edit