

Catedra de Calculatoare

## **Documentatie Proiect PG**

Popa Alexandra-Maria  
Grupa 30234

16 ianuarie 2020

# Cuprins

<b>1</b>	<b>Prezentarea temei</b>	<b>3</b>
<b>2</b>	<b>Scenariul</b>	<b>4</b>
2.1	Descrierea scenei si a obiectelor . . . . .	4
2.2	Functionalitati . . . . .	4
<b>3</b>	<b>Detalii de implementare</b>	<b>5</b>
3.1	Functii si algoritmi . . . . .	5
<b>4</b>	<b>Prezentarea interfetei grafice utilizator/manual de utilizare</b>	<b>7</b>
<b>5</b>	<b>Concluzii si dezvoltari ulterioare</b>	<b>9</b>
	<b>Bibliografie</b>	<b>10</b>

# 1 Prezentarea temei

Proiectul ales prezinta o scena de obiecte 3D din natura ce are ca scop crearea peisajului fotorealist a unui parc. Am ales tema unui parc deoarece in acest mod am reusit sa aduc impreuna elementele urbane cu cele din natura si sa creez o reprezentare cat se poate de clara a acestora prin intermediul librariilor prezentate in laborator.

## 2 Scenariul

### 2.1 Descrierea scenei si a obiectelor

Scena creata reprezinta un parc format din banci, lampi de iluminat stradal, cosuri de gunoi, un foisor, leagan de joaca pentru copii, toate acestea fiind inconjurate de copaci si tufisuri. De asemenea, parcul este amplasat intr-un cartier, astfel se pot observa prezenta caselor de la margine. Obiectele alese au fost pozitionate astfel incat sa creeze un cadru restrans din realitate, astfel a fost construita o alea pe care se afla de o parte si de alta a sa banci si elemente specifice, precum si copaci dar si locuri de relaxare.

### 2.2 Functionalitati

Vizualizarea scenei de obiecte este posibila datorita miscarilor urmatoare:

- ⇒ sus
- ⇒ jos
- ⇒ stanga
- ⇒ dreapta
- ⇒ rotatie la stanga in jurul axei Y
- ⇒ rotatie la dreapta in jurul axei Y

Totodata scena poate fi vizualizata in mai multe forme: solid, wireframe si point( G,F,H) Efaptul de ceata poate fi activat apasand tasta C si dezactivat, apasand tasta V.

# 3 Detalii de implementare

Asezarea obiectelor in scena a fost posibila prin operatii de translatie, scalare si rotatie. Obiectele au fost texturate utilizand Blender 2.9 [5], iar mai apoi folosite in crearea scenei. Aceste obiecte au fost descarcate, apoi texturate si aplicate diferite transformari asupra lor.[4][1][3]

## 3.1 Functii si algoritmi

Pentru realizarea proiectului au fost create functii utilizand limbajul de programare C++ si au fost folosite atat functii si headere din OpenGL, cat si functii noi necesare pentru formarea camerei, a modelelor sau a efectelor. Totodata au fost create shadere pentru desenarea si incarcarea obiectelor.[2] Printre principalele functii utilizate, se regasesc;

initOpenGLWindow() ⇒ initializeaza fereastra de vizualizare OpenGL  
initOpenGLState() ⇒ initializare inainte de crearea scenei  
initModels() ⇒ initializare modele 3D ce vor aparea in scena  
initShaders() ⇒ initializare si incarcare shadere  
initSkyBox() ⇒ initializare SkyBox cu imaginile care il formeaza  
initUniforms() ⇒ initializare variabile uniform  
initFramebuffer() ⇒ initializare FrameBuffer  
renderScene() ⇒ functia in care se formeaza scena de obiecte( tot aici se produc transformari pe obiecte)  
processMovement() ⇒ functie utilizata in renderScene() pentru identificarea functiilor anumitor taste  
light\_rds() ⇒ functie folosita pentru crearea umbrelor  
updateDelta1(double elapsedSeconds) ⇒ functie utilizata pentru crearea animatiei legata de pasari

Au fost utilizate numeroase shadere pentru a calcula componentele luminii, efectul de ceata, dar si umbrele.[2]

Efectul de ceata a fost creat prin intermediul modelui exponential patratice, astfel un rezultat mai bun poate fi obtinut luand in considerare reducerea intensitatii luminii in functie de distanta. Factorul de atenuare care va fi folosit reprezinta densitatea de ceata, aceasta densitate fiind constanta in toata scena. Rezultatul este o scadere rapida a factorului de ceata in comparație cu abordarea liniara.

Lumina si culorile regasite in proiect sunt formate prin calcularea componentelor luminii in fragment shader. Astfel cele trei componente ale luminii ambientala, difusa si speculara se calculeaza astfel:

```
//compute ambient light
ambient = ambientStrength * lightColor;

//compute diffuse light
diffuse = max(dot(normalEye, lightDirN), 0.0f) * lightColor;
```

```
//compute specular light
float specCoeff = pow(max(dot(halfVector, normalEye), 0.0f), shininess);
specular = specularStrength * specCoeff * lightColor;
```

## **4 Prezentarea interfetei grafice utilizator/manual de utilizare**

Proiectul se deschide utilizand VisualStudio 2015 si se ruleaza, urmand mai apoi sa apara scena in care se poate naviga folosind miscarile camerei de vizualizare:

A ⇒ miscare la stanga  
D ⇒ miscare la dreapta  
W ⇒ miscare de apropiere  
S ⇒ miscare de departare  
Q ⇒ roatire in jurul lui Y la stanga  
E ⇒ roatire in jurul lui Y la dreapta  
G ⇒ activare mod solid  
F ⇒ activare mod wireframe  
P ⇒ activare mod point  
C ⇒ activare efect de ceata  
V ⇒ dezactivare efect de ceata  
UP ⇒ dezactivare efect ziua  
DOWN ⇒ dezactivare efect noapte



Figure 4.1:



Figure 4.2:



## **5 Concluzii si dezvoltari ulterioare**

Proiectul creat incerca sa prezinta o scena cat mai realista a unui parc cu tot ceea ce tine de acesta, insemnand obiecte specifice sau efecte.

Ulterior proiectul poate fi dezvoltat prin adaugarea de surse multimplle de lumina, crearea unui mod de noapte si zi, efecte de ploaie sau animatii de obiecte mai complexe.

# Bibliografie

- [1] *Free3D*. URL: <https://free3d.com/>.
- [2] *Laborator SPG*. URL: <https://moodle.cs.utcluj.ro/course/view.php?id=186>.
- [3] *SketchFab*. URL: <https://sketchfab.com/feed>.
- [4] *Turbosquid*. URL: <https://www.turbosquid.com/>.
- [5] *Tutorial Blender*. URL: <https://moodle.cs.utcluj.ro/course/view.php?id=186>.