



PROIECT PRELUCRARE GRAFICĂ

Ghiorghioiu Denisa Elena

Grupa 30234 Seria A Calculatoare Română

Curpins

Cuprins	2
2.Prezentarea temei	3
3. Scenariul.....	3
3.1 Descrierea scenei si a obiectelor.....	3
3.2 Funcționalități.....	4
4.Detalii de implementare	4
4.1 Funcții și algoritmi	4
4.1.1 Soluții alese	4
4.1.2 Motivarea abordării alese.	7
4.2 Modelul grafic	7
4.3 Structuri de date.....	7
4.4 Ierarhia de clase.....	9
5.Prezentarea interfeței grafice utilizator/ manual de utilizare	9
6.Concluzii și dezvoltări ulterioare.....	9
7.Referințe	9

2. Prezentarea temei

Am realizat acest proiect în cadrul cursului de Prelucrare Grafică și are ca obiectiv principal dezvoltarea unei scene fotorealiste utilizând cunoștințele despre OpenGL. Utilizatorii pot explora scena folosind tastele și mouse-ul. Scena pune în evidență efecte precum iluminare, ceață, texturare, ploaie, mișcare pentru a oferi o soluție cât se poate de realistă.

3. Scenariul

3.1 Descrierea scenei si a obiectelor

Scena ilustrează tipologia vieții la sat. În mijlocul satului se află o biserică, înconjurată de drum pietros. În jurul bisericii sunt căsuțe din lemn, iar lângă acestea se mai află și un hambar, un grajd. Totodată, la marginea scenei avem și un lac mic pe care se mișcă o barcă. În scenă am inclus și copaci, flori, animale într-un Țarc și o moară de vânt. Am realizat această scenă cu gândul la copilărie, când totul era fără griji.





3.2 Funcționalități

În scenă am introdus multe funcționalități precum:

- W,A,S,D permite mișcarea camerei
- Tastele L și N permit vizualizarea scenei atât ziua cât și noaptea
- Tasta P pornește/oprește ploaia
- Tasta F activează/dezactivează ceața
- Tastele 1,2,3,4 permit vizualizarea scenei în modurile solid, wireframe, poligonal și smooth
- Tasta H activează lumina direcțională(soarele)
- Tasta J activează luminile spot(felinarele de lângă biserică)
- Tasta K activează luminile punctiforme(lumineaza casele pe întuneric)
- Tasta V activează modul de mișcare automată a camerei prin scenă
- Tasta M care activează/dezactivează rotirea morii de vânt

4.Detalii de implementare

4.1 Funcții și algoritmi

4.1.1 Soluții alese

În implementarea proiectului, am folosit funcții și algoritmi pentru implementarea unor funcționalități cum ar fi:

- Pentru a genera efectul de ceață, am folosit formula pentru ceață exponențială pătratică, unde $\text{fragmentDistance} * \text{fogDensity}$ determină cât de mult influențează distanța și densitatea ceții.

$$\text{fogFactor} = e^{-(\text{fragmentDistance} * \text{fogDensity})^2}$$

M-am folosit și de funcția **clamp()** care îți oferă o valoare pentru fogFactor între 0 și 1. Dacă e 0 înseamnă că avem o ceață foarte densă, iar dacă este 1, obiectul este complet vizibil. Am activat ceața prin intermediul unei variabile uniform adăugată în shader, care va primi valoarea true atunci când tasta F este apăsată.



- Pentru a genera efectul de ploaie, am generat un set de picături de ploaie, fiecare cu o poziție aleatorie și o viteză. Pentru a genera poziția aleatorie am folosit funcția `rand()`. Toate picăturile le-am stocat într-un vector **raindrops**. Dacă o picătură ajunge sub nivelul solului, poziția sa este resetată. Am activat ploaia prin intermediul unei variabile, care devine true atunci când tasta P este apăsată. Dacă tasta este apăsată, se va randa ploaia.



- Mișcarea morii de vânt: m-am folosit de funcționalitățile Blender-ului, și am despărțit obiectul în 2 obiecte, după care moara de vânt introdus-o în scenă ca un obiect static, și morișile le-am pus separat, ca să efectuez și rotații în jurul axei Z. Acestea se vor mișca doar după ce voi apăsa tasta M.
- Mișcarea bărcii este o mișcare continuă pe lac, aceasta începe să se miște odată cu rularea scenei.
- Luminile direcționale sunt surse de lumină infinit de îndepărtate. Toate razele de lumină care provin dintr-o sursă de lumină direcțională sunt paralele și astfel direcția luminii implicate în calculul iluminării este constantă între obiecte și toate fragmentele acestora. Astfel, aceste surse de lumină sunt ideale pentru iluminarea globală din cadrul scenelor noastre. La mine în scenă, lumina direcțională am ales-o ca fiind soarele, deoarece ea îmi luminează toată scena. Am utilizat cele 3 componente ale luminii ambient (iluminează obiectele indiferent de poziție), diffuse (lumina care depinde de unghiul dintre suprafață și direcția luminii), specular (reflexiile).
- Luminile punctiforme sunt surse de lumină cu o poziție dată care iluminează radial și uniform în toate direcțiile. Spre deosebire de luminile direcționale, razele generate de luminile punctiforme se estompează în funcție de distanță, făcând astfel obiectele mai apropiate de sursă să pară mai iluminate decât obiectele mai îndepărtate. Luminile punctiforme le-am evidențiat prin iluminarea unor case. Am folosit tot cele 3 componente ale luminii, însă am adăugat și atenuarea luminii care reduce intensitatea luminii pe măsură ce distanța crește. Atenuarea este invers proporțională cu constant, linear (scade liniar cu distanța) și quadratic (scade exponențial cu distanța). La final, la calculul luminii, aplicăm atenuarea:

ambient *= attenuation; diffuse *= attenuation; specular *= attenuation;



- Luminile spotlight sunt un tip de iluminare utilizată pentru a evidenția sau accentua un anumit obiect, zonă sau persoană. Ele emit un fascicul de lumină concentrat, cu margini clare, care poate fi reglat pentru a controla intensitatea și direcția luminii. Lumina spot am evidențiat-o prin utilizarea a 2 felinare lângă biserică pentru a fi luminată. Pentru calculul luminii spotlight folosim în plus atenuarea și distanța (distanța dintre fragment și lumină), theta care reprezintă produsul scalar dintre direcția luminii și direcția centrală de unde pornește lumina spot, epsilon care e diferența dintre inCutOff (unghiul interior) și outCutOff (unghiul exterior) și intensitatea care e maximă în centru și scade spre margini.



4.1.2 Motivarea abordării alese

Am ales această abordare, deoarece am înțeles și la laborator cum să implementez și mi s-au părut implementări importante pentru a reda o scenă cât mai realistă.

4.2 Modelul grafic

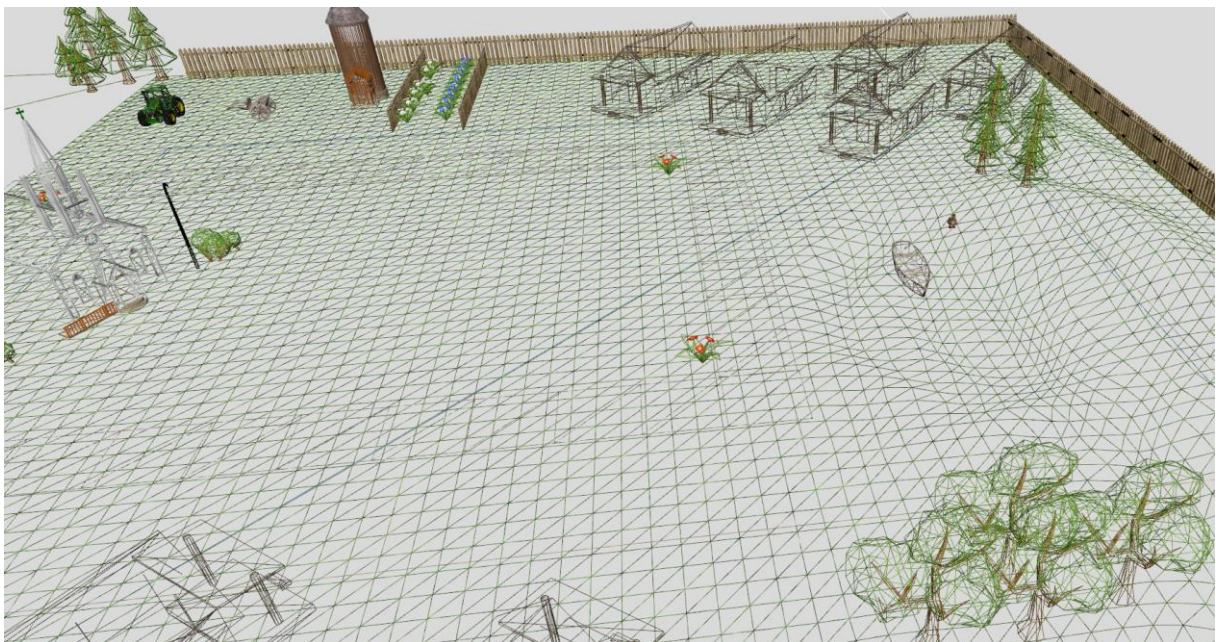
Am implementat scena folosind Blender, importând obiectele statice în scenă, după care am importat obiectele dinamice(moriștile, barca) și picătura de ploaie(desenată în Blender) în OpenGL. Am adăugat și 2 skybox-uri(unul pentru zi și unul pentru noapte) pentru a crea iluzia unui spațiu în jurul scenei.

Prezentarea celor 3 moduri de vizualizare:

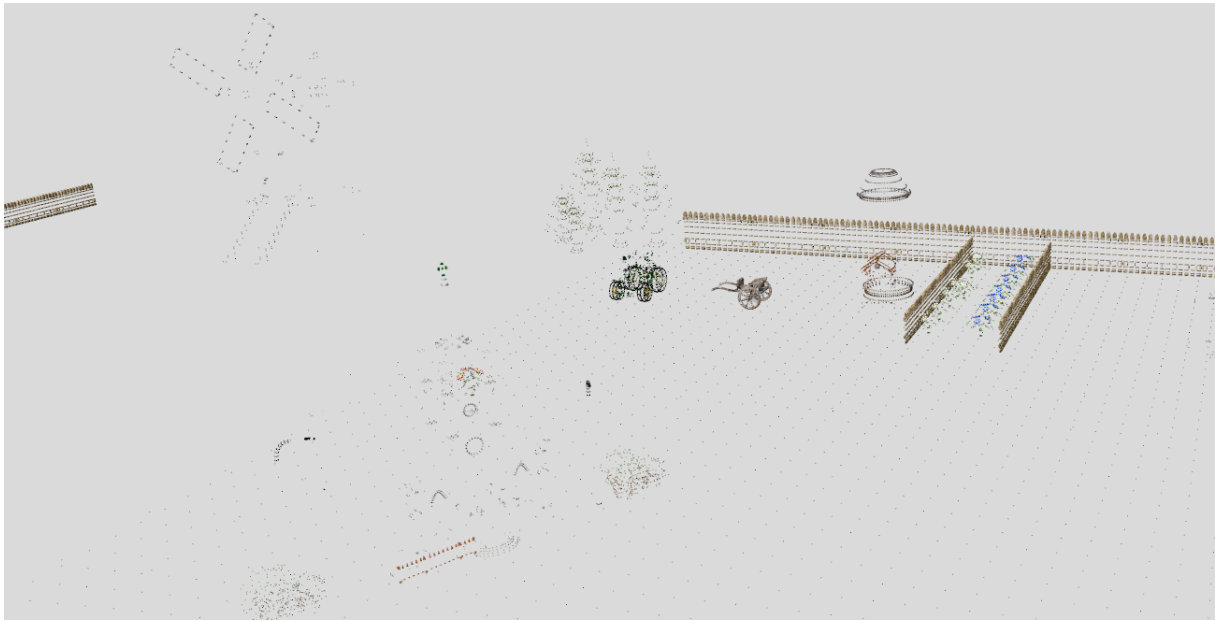
SOLID:



WIREFRAME:



POLYGONAL:



4.3 Structuri de date

```
// window
gps::Window myWindow;

// matrices
glm::mat4 model;
glm::mat4 skiModel;
glm::mat4 houseModel;
glm::mat4 view;
glm::mat4 projection;
glm::mat3 normalMatrix;

// light parameters
glm::vec3 lightDir;
glm::vec3 lightColor;

// shader uniform locations
GLint modelLoc;
GLint viewLoc;
GLint projectionLoc;
GLint normalMatrixLoc;
GLint lightDirLoc;
GLint lightColorLoc;

// camera
gps::Camera myCamera(
    glm::vec3(0.0f, 10.0f, 3.0f),
    glm::vec3(0.0f, 0.0f, -10.0f),
    glm::vec3(0.0f, 1.0f, 0.0f));

GLfloat cameraSpeed = 1.0f;

GLboolean pressedKeys[1024];

//scene
```

```
gps::Model3D landscape;  
glm::mat4 landscapeModel;  
glm::mat3 landscapeNormalMatrix;
```

```
//windmill  
gps::Model3D windmill;  
glm::mat4 windmillModel;  
glm::mat3 windmillNormalMatrix;
```

```
//rain  
gps::Model3D raindropModel;  
struct Raindrop {  
    glm::vec3 position;  
    float speed;  
};
```

```
std::vector<Raindrop> raindrops;  
const int maxRaindrops = 5000;
```

```
GLfloat angle;
```

```
//boat  
gps::Model3D boat;  
glm::mat4 boatModel;  
glm::mat3 boatNormalMatrix;  
bool boatMovingRight = true;  
float boatSpeed = 0.05f;
```

```
// shaders  
gps::Shader myBasicShader;
```

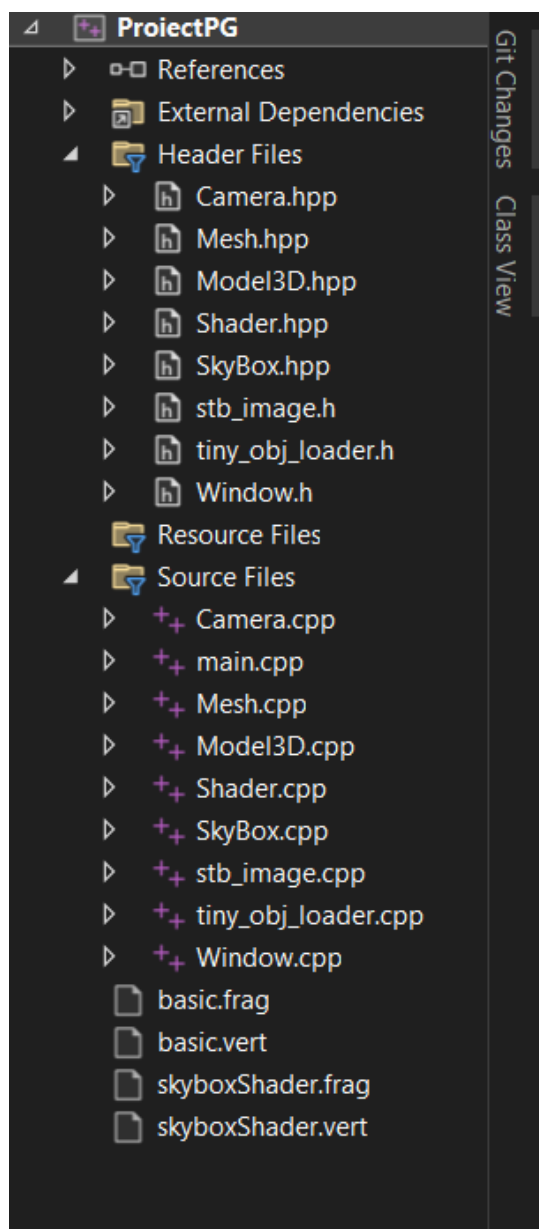
```
gps::SkyBox mySkyBox;  
gps::SkyBox mySkyBoxNight;
```

```
gps::Shader skyboxShader;
```

```
//render mode  
enum RenderMode { SOLID, WIREFRAME, POLYGONAL, SMOOTH };  
RenderMode currentRenderMode = SOLID;
```

```
//automatic tour  
std::vector<glm::vec3> tourPoints = {  
    glm::vec3(56.8058f, 8.64644f, - 109.115f),  
    glm::vec3(- 136.705f, 13.5886f, - 150.389f),  
    glm::vec3(- 298.978f, 19.3819f, - 184.88f),  
    glm::vec3(- 373.986f, 21.6437f, - 107.209f),  
    glm::vec3(- 273.238f, 27.5056f, - 42.2815f),  
    glm::vec3(- 125.468f, 27.5056f, - 50.5429f);}
```

4.4 Ierarhia de clase



5. Prezentarea interfeței grafice utilizator/ manual de utilizare

Rularea aplicației se va face din ProiectPG.exe. Pentru a naviga prin scenă, vom avea nevoie doar de tastatură (tastele specificate mai sus) și de mouse. Pentru a activa lumina în scenă trebuie să apăsați tasta H, deoarece inițial lumina direcțională nu este activată. Se poate naviga în orice colț al scenei.

6. Concluzii și dezvoltări ulterioare

Pentru mine acest proiect a fost un proiect interesant, care m-a făcut să înțeleg mai bine cum funcționează Blender și OpenGL. Am învățat multe lucruri noi, pentru mine adăugarea luminilor punctiforme și spot fiind puțin mai dificil.

Pentru a face scena mult mai interesantă, și mai realistă, se pot adăuga umbre, vânt, sunete, transparență, detectarea coliziunilor și mai multe obiecte să fie în mișcare.

7. Referințe

- Laboratoarele de la prelucrare grafică
- <https://learnopengl.com/Lighting/Multiple-lights>
- <https://learnopengl.com/Lighting/Light-casters>
- Tutoriale YouTube:
https://www.youtube.com/watch?v=ckGg7aUGoPQ&list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM&index=1