

Procesare Grafica

- Scena 3D a unui oras medieval -

Spinean Sebastian

15/01/2021

Contents

1 Prezentarea temei	3
2 Scenariu	4
2.1 Descrierea scenei si a obiectelor	4
2.2 Functionalitati	4
3 Detalii de implementare	11
3.1 Functii si algoritmi	11
3.1.1 Animatia scenei	11
3.1.2 Animatia elicei morii de vant	11
3.1.3 Animatia carului	11
3.1.4 Ploaia	11
3.1.5 Fulger	11
3.1.6 Detectia colizionilor	11
3.2 Modelul grafic	11
3.3 Structuri de date	12
3.4 Ierarhia de clase	12
4 Manual de utilizare	13
5 Concluzii si dezvoltari ulterioare	14
6 Referinte	15
6.1 Obiecte	15
6.2 Altele	15

1 Prezentarea temei

Proiectul a presupus realizarea unei scene 3D foto-realiste folosind OpenGL. De asemenea, proiectul implementeaza o serie de efecte pentru a creste realismul scenei precum: generarea umbrelor, efectul de ceata, animatii, efectul de ploaie, fulger si detectia coliziunilor

2 Scenariu

2.1 Descrierea scenei si a obiectelor

Scena prezinta un oras medieval imprejmuit de ziduri. Orasul se afla in centrul scenei. De asemenea, in scena mai exista un lac pe malul caruia se afla o moara de vant si doua hambare. Orasul se leaga de lac printr-un drum.

Majoritatea obiectelor au fost exportate sub forma unui singur obj mai putin obiectele pentru care s-au implementat animatii.

Principalele obiecte 3d folosite sunt:

- ground plane
- road cube
- skydome
- *Biserica*
- *Capela*
- *Casa 1*
- *Casa 2*
- *Casa 3*
- *Casa 4*
- *Hambar*
- *Moara*

2.2 Functionalitati

Proiectul cuprinde urmatoarele functionalitati:

- Vizualizarea scenei prin deplasarea camerei in fata, spate stanga si dreapta utilizand tastele **W**, **A**, **S**, **D**
- Rotirea camerei utilizand mouse-ul
- Animatia scenei de la tasta **T**
- Activarea si dezactivarea lumintii directionale de la tasta **L**
- Vizualizarea in mod wireframe de la tasta **N** si revenirea in modul solid de la tasta **M**
- Activarea modului smooth de la tasta **B** si dezactivarea lui de la tasta **V**
- Vizualizarea umbrelor
- Animatia elicei de la moara de vant de la tasta **R**
- Animatia unui car de la tasta **Y**
- Activarea efectului de cata **C**
- Activarea ploii si a fulgerului de la tasta **P**
- Detectia coliziunilor pentru moara de vant si hambarele de langa lac.

Imagini prezentand scena si efectele implementate

Figure 1: Scena

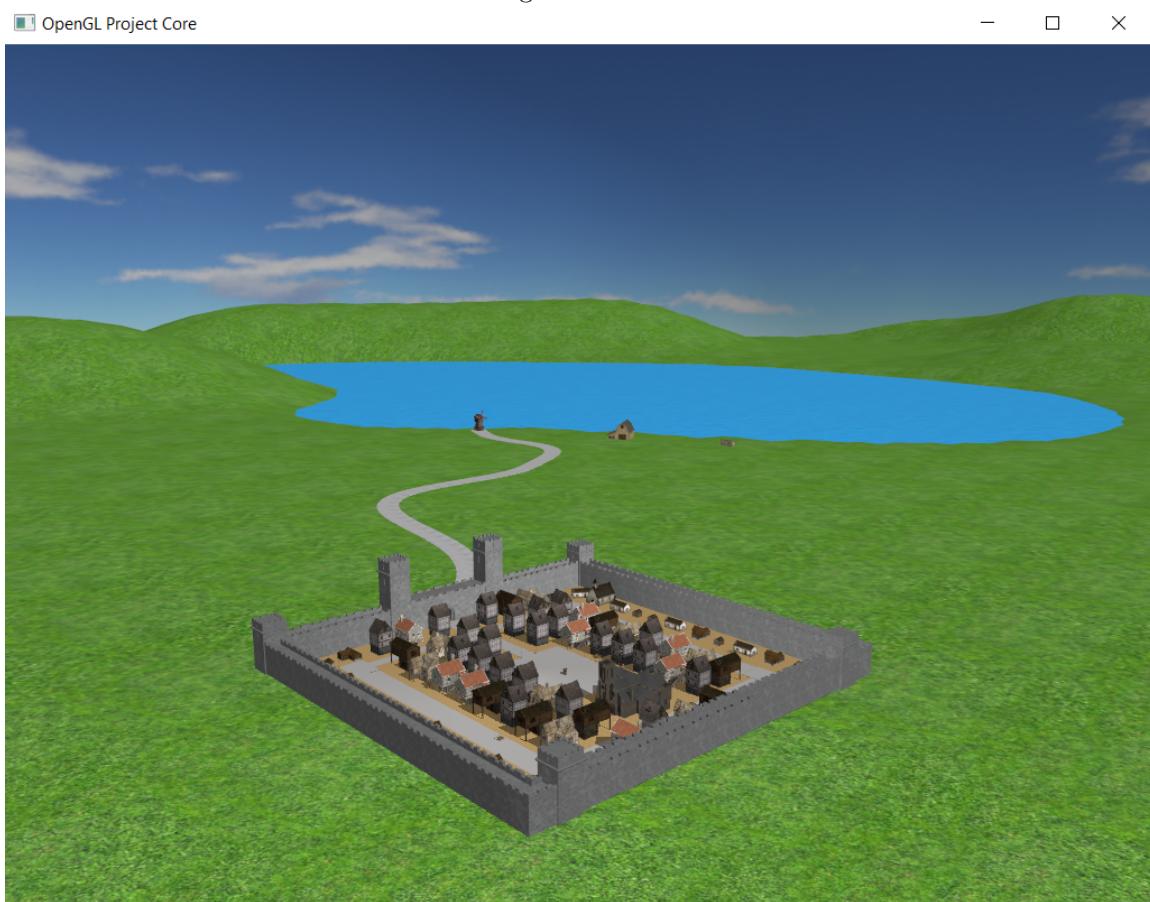


Figure 2: Lumina punctiforma

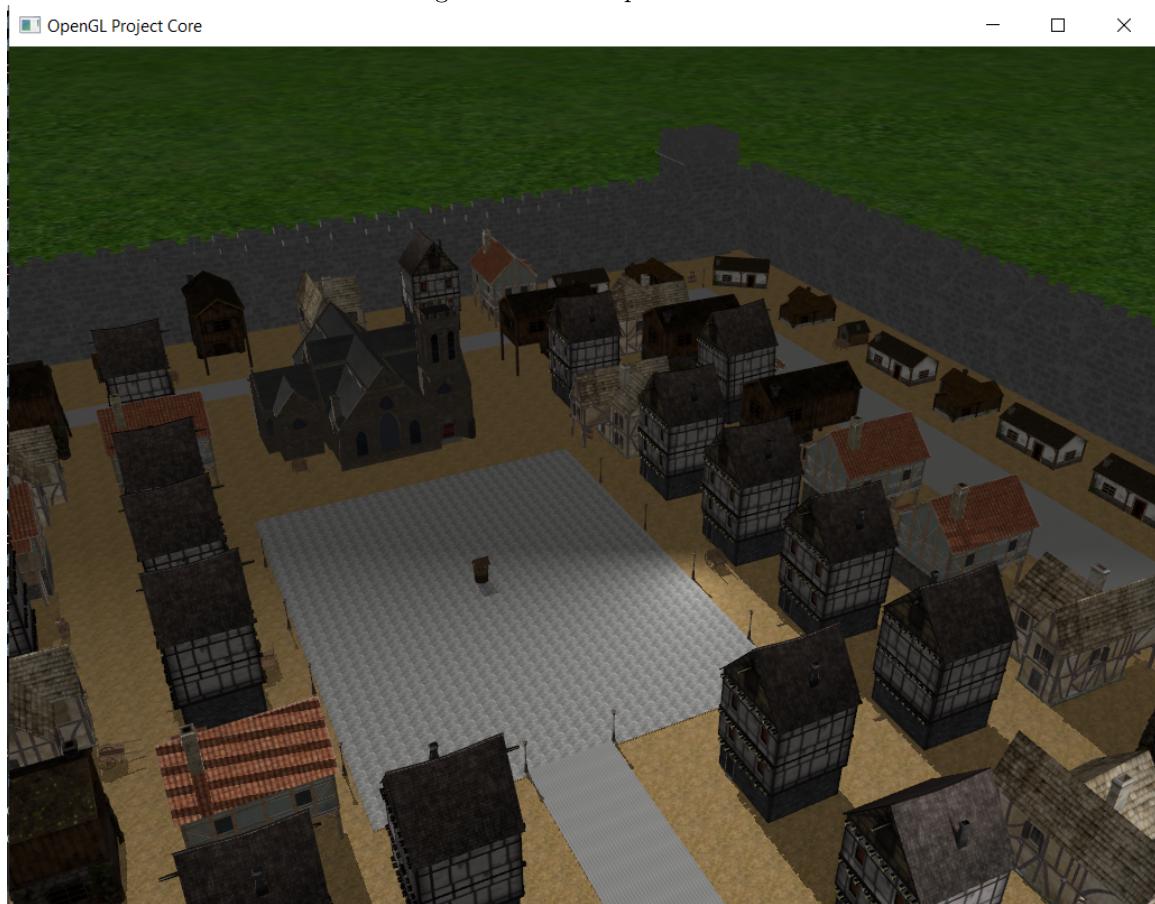


Figure 3: Umbre

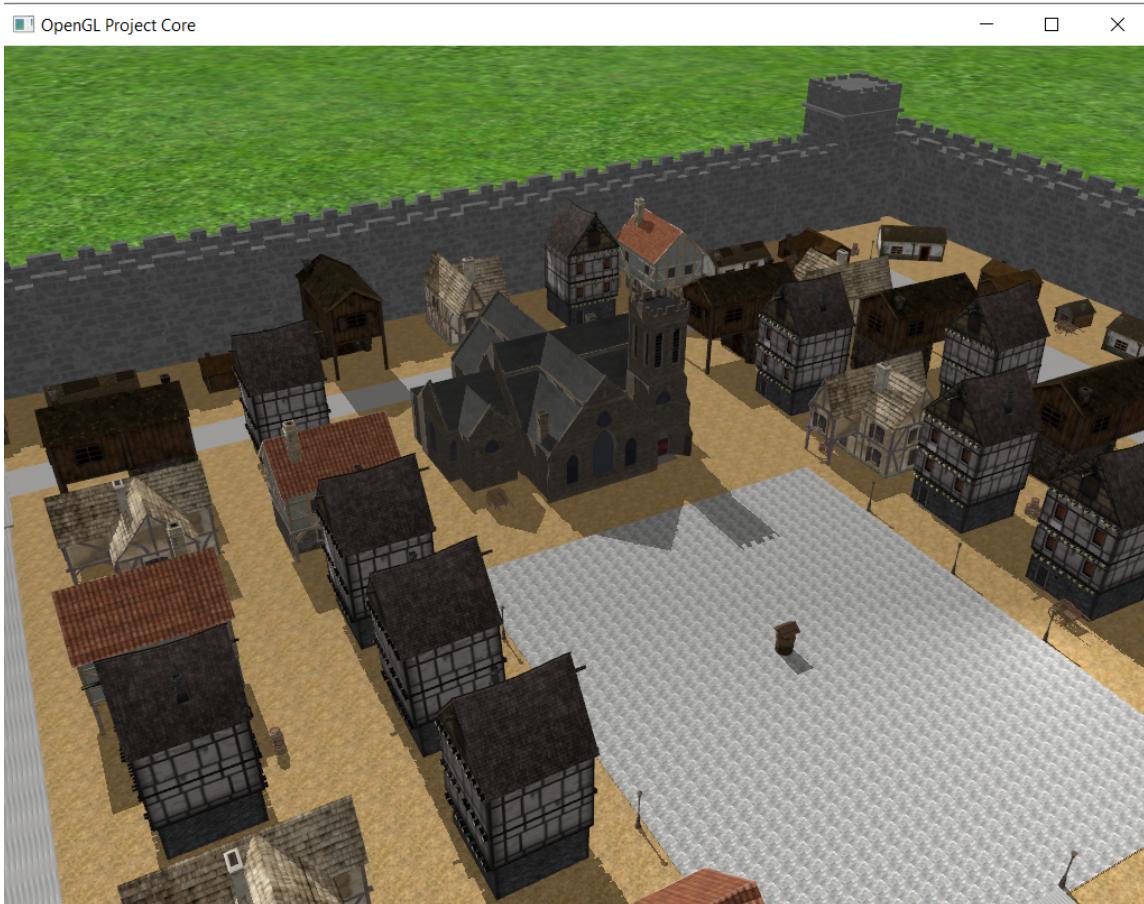


Figure 4: Ceata

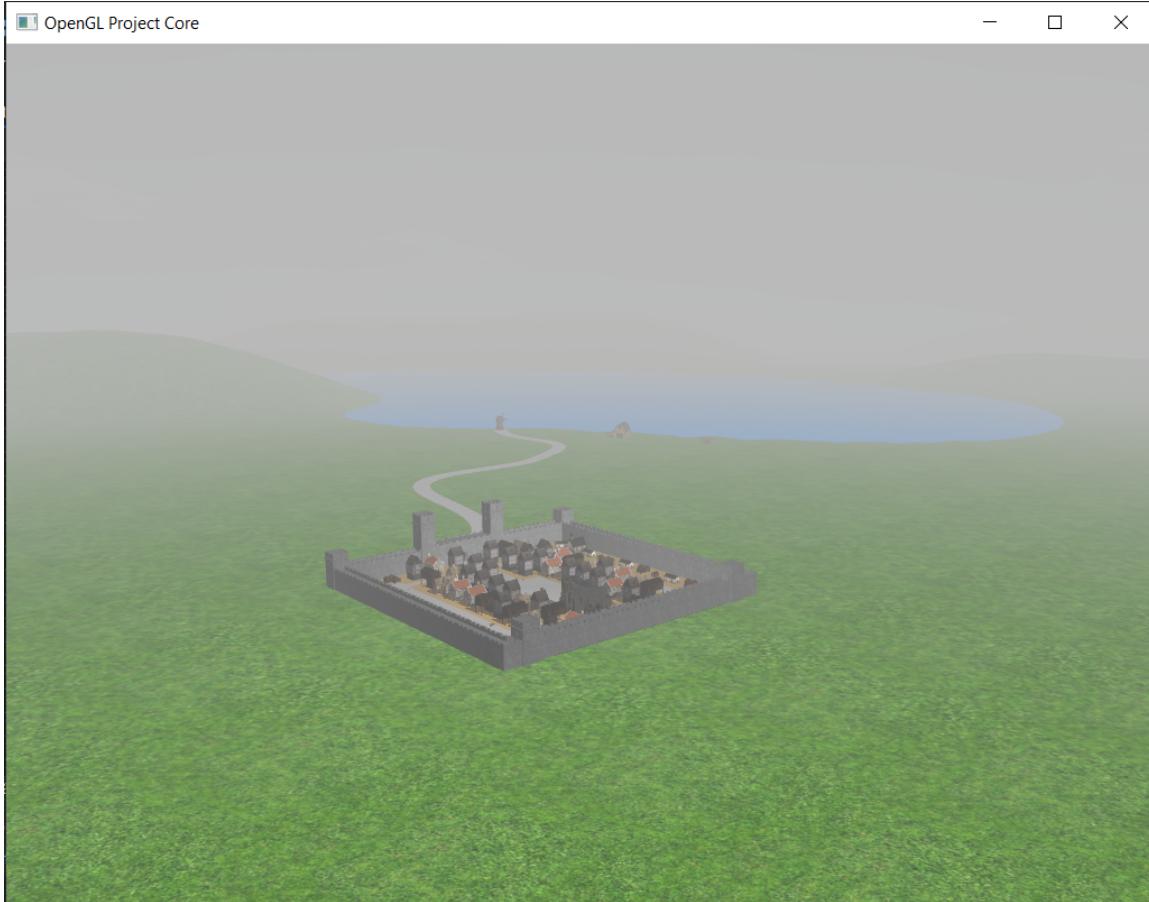


Figure 5: Ploaie

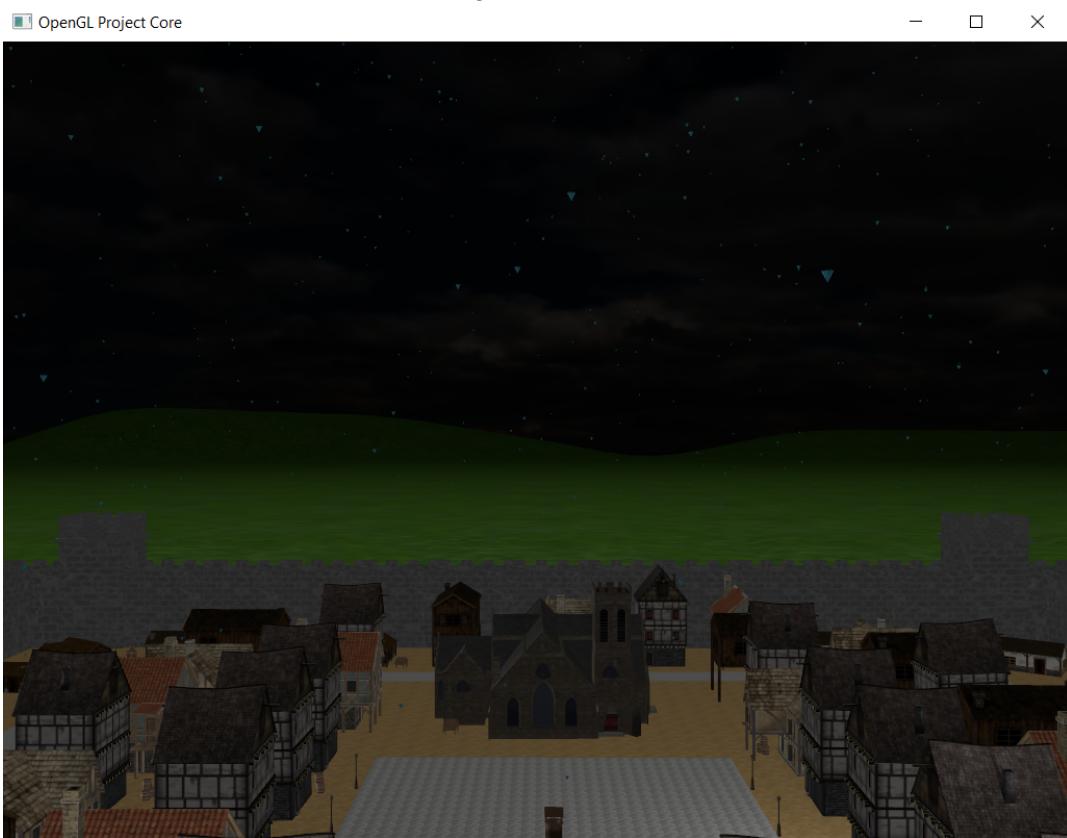
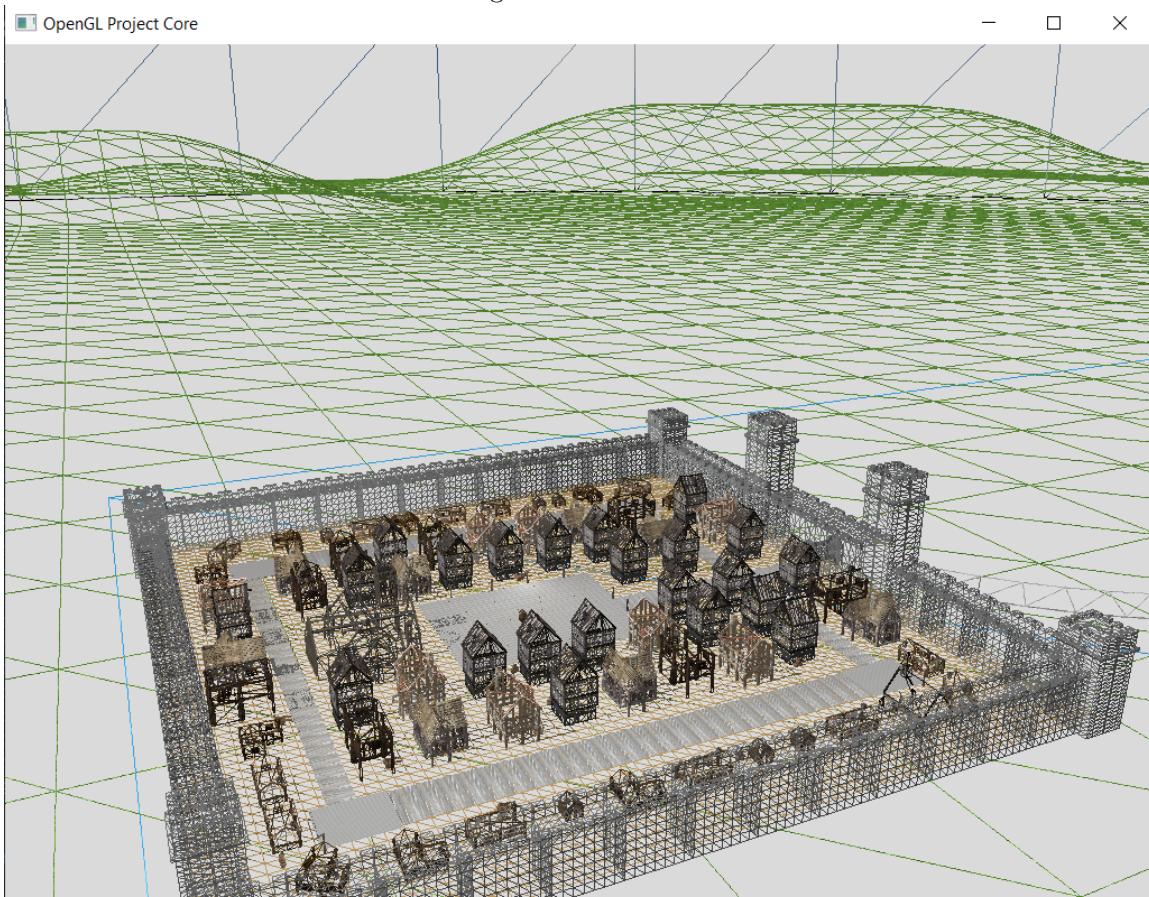


Figure 6: Wireframe



3 Detalii de implementare

3.1 Functii si algoritmi

Pentru a creste gradul de foto-realism s-au implementat mai multi algoritmi pentru diferite efecte.

3.1.1 Animatia scenei

Animatia scenei presupune deplasarea camerei pentru a se vizualiza aspectele importante ale scenei. Pentru asta s-a folosit un. Camera se deplaseaza de-a lungul laturilor orasului. Pentru asta s-au folosit patru liste ce contin elemente de tip vec3 reprezentand pozitiile prin care trebuie sa treaca camera. Fiecare lista coresponde uneia dintre laturi.

3.1.2 Animatia elicei morii de vant

Pentru a realiza aceasta animatie elicea a trebuit sa fie un obj diferit de restul morii. La apasarea tastei R variabila delta care reprezinta unghiul de rotatie se incrementeaza. Elica se afla in punctul de coordonate (0 ,0 ,0). Prima data se executa o rotatie cu unghiul delta fata de axa z dupa care se face o translatie in pozitia finala pe care ar trebui sa o aiba elica.

3.1.3 Animatia carului

Pentru aceasta animatie corpul carului, roata din stanga si roata din dreapta au trebuit sa fie obj-uri diferite. Carul se deplaseaza la apasarea tastei **Y**. Aceasta merge in fata pana cand coordonata Z atinge valoarea - 1.2f dupa care merge cu spatele pana in pozitia initiala. Pentru corpul carului se realizeaza o translatie, coordonata Z modificandu-se la apasarea tastei **Y**. Pentru cele doua roti se realizeaza o rotatie cu un unghi numit rotireRoata dupa axa X urmand o translatie, coordonata Z modificandu-se de fiecare data.

3.1.4 Ploaia

Pentru realizarea ploii s-au rasterizat 1000 de picuri de ploaie. Pozitia fiecaruia s-a determinat in mod aleatoriu. Alegerea pozitiei pentru fiecare picure s-a realizat in functia main. Asupra fiecarui picure de ploaie se realizeaza o translatie pe axa Z.

3.1.5 Fulger

Pentru a implementa fulgerul s-a folosit o variabila care este incrementata la desenarea scenei de fiecare data cand ploaia este activata. Variabila este trimisa la shader si acolo are loc o luminare a scenei cand variabila are o valoare intre 980 si 1000. Cand ajunge la valoarea 1000 se reseteaza la 0.

3.1.6 Detectia coliziunilor

Detectia coliziunilor s-a realizat pentru moara de vant si pentru cele doua hambare de langa lac. Pentru asta fiecare obiect a fost incadrat intr-o sfara. La fiecare deplasare a camerei prin scena se verifica daca pozitia camerei este sau nu in afara unei sfere. Pentru a determina asta se utilizeaza ecuatia sferei si se verifica daca valoarea obtinuta pentru pozitia camerei este mai mica decat raza sferei. In caz afirmativ se semnaleaza o coliziune si camera nu mai poate inainta.

3.2 Modelul grafic

Modelul grafic folosit se bazeaza pe modelul de ShadowMapping, modelul Phong.

3.3 Structuri de date

S-au folosit structurile de date specifice bibliotecii GLM cum ar fi `vec<n>` si `mat<n>`. De asemenea, s-au folosit structuri din OpenGL precum `GLuint`.

3.4 Ierarhia de clase

Proiectul cuprinde urmatoarea ierarhie de clase:

- Camera.hpp - defineste camera si miscariile ei
- Mesh.hpp - defineste varfurile unui obiect
- Model3D.hpp - defineste un obiect
- Shader.hpp - defineste un shader

4 Manual de utilizare

Pentru a putea folosi aplicatia trebuie sa:

1. Sa se ruleze fisierul executabil
2. Sa se utilizeze comenziile prezentate in sectiunea Functionalitati pentru a vedea diferitele efecte

5 Concluzii si dezvoltari ulterioare

In concluzie, proiectul prezinta o scena 3D realista a unui oras medieval. La cresterea foto-realismului au contribuit efectele implementate precum: umbre, ceata, animatii, polaie si fulger.

Pentru dezvoltari ulterioare s-ar putea implementa:

- Detectia coliziunilor pentru toate obiectele
- Umbre pentru lumina punctiforma
- Introducerea de sunete

6 Referinte

6.1 Obiecte

1. *Biserica* - <https://sketchfab.com/3d-models/gethsemane-episcopal-church-83721522046f4a6f8c3fb7f26d26d231>
2. *Casa 1* - <https://www.turbosquid.com/3d-models/free-medieval-house-3d-model/727769>
3. *Casa 2* - <https://www.turbosquid.com/3d-models/medieval-house-3d-model/809615>
4. *Casa 3* - <https://sketchfab.com/3d-models/medieval-house-4ec56df1c24d422ea85d1cfbf21bbe8c>
5. *Casa 4* - <https://sketchfab.com/3d-models/medieval-house-4-f59b5a688aa64da6b2d97201b0cc05c9>
6. *Capela* - <https://www.turbosquid.com/3d-models/free-chapel-celestial-video-3d-model/464431>
7. *Hambar* - <https://sketchfab.com/3d-models/barn-70eeba9601a047c881abcd9d40eb26e>
8. *Moara* - <https://www.turbosquid.com/3d-models/3d-model-windmill-simple/1049876>

6.2 Altele

1. *Lumini* - <https://learnopengl.com/Lighting/Multiple-lights>
2. *Umbre* - <https://learnopengl.com/Advanced-Lighting/Shadows/Shadow-Mapping>
3. *Camera* - <https://learnopengl.com/Getting-started/Camera>
4. *Anti-aliasing* - <https://learnopengl.com/Advanced-OpenGL/Anti-Aliasing>