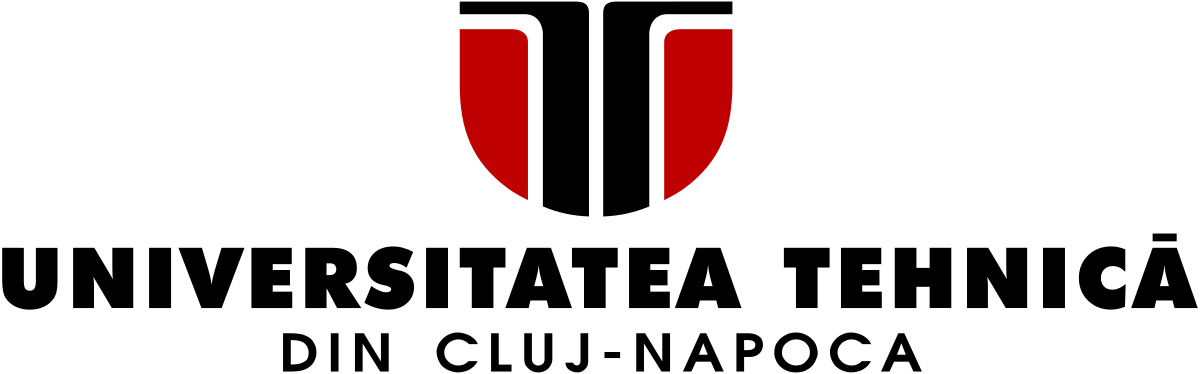
**Universitatea Tehnica din Cluj Napoca**

**Calculatoare și Tehnologia Informației**

Proiect Prelucrare Grafica



Varvari Alexandru  
30235

Prof. îndrumător Mihon Danut

**-Cuprins -**

[1.Introducere 3](#_Toc156451703)

[Contextul temei 3](#_Toc156451704)

[2. Scenariu 3](#_Toc156451705)

[2.1 Descriere scenă și obiecte 3](#_Toc156451706)

[2.2 Funcționalitati 4](#_Toc156451707)

[3. Implementare 7](#_Toc156451708)

[3.1 Funcții și algoritmi 7](#_Toc156451709)

[3.2. Modelul grafic 10](#_Toc156451710)

3.3 Structuri de date.......................................................................................................................11

[3.4. Ierarhia de clase 12](#_Toc156451711)

[5. Interfața utilizator 13](#_Toc156451712)

[6. Concluzii si dezvoltari ulterioare 13](#_Toc156451713)

[7. Referințe 14](#_Toc156451714)

# 

# 1.Introducere

## Contextul temei

Scopul principal al acestui proiect este de a aprofunda cunoștințele legate de realismul vizual în jocurile video și de a înțelege modul în care un computer generează și redă o scenă plină de obiecte diverse. In ceea ce priveste constructia scenei am optat in crearea unei scene low-poly pentru simplitatea obiectelor folosite si utilizarea unui peisaj cartoon based.

# 2. Scenariu

## 2.1 Descriere scenă și obiecte

În prim-plan, pe partea stângă, se află o biserică cu arhitectură tradițională și un acoperiș înclinat, lângă care este plasată o bancă și o fântână manuală. În centrul imaginii, un pod din lemn traversează un curs de apă. Dupa acel pod se afla o moara de vant, in imprejurimea careia putem vedea mai multe casute din sat.



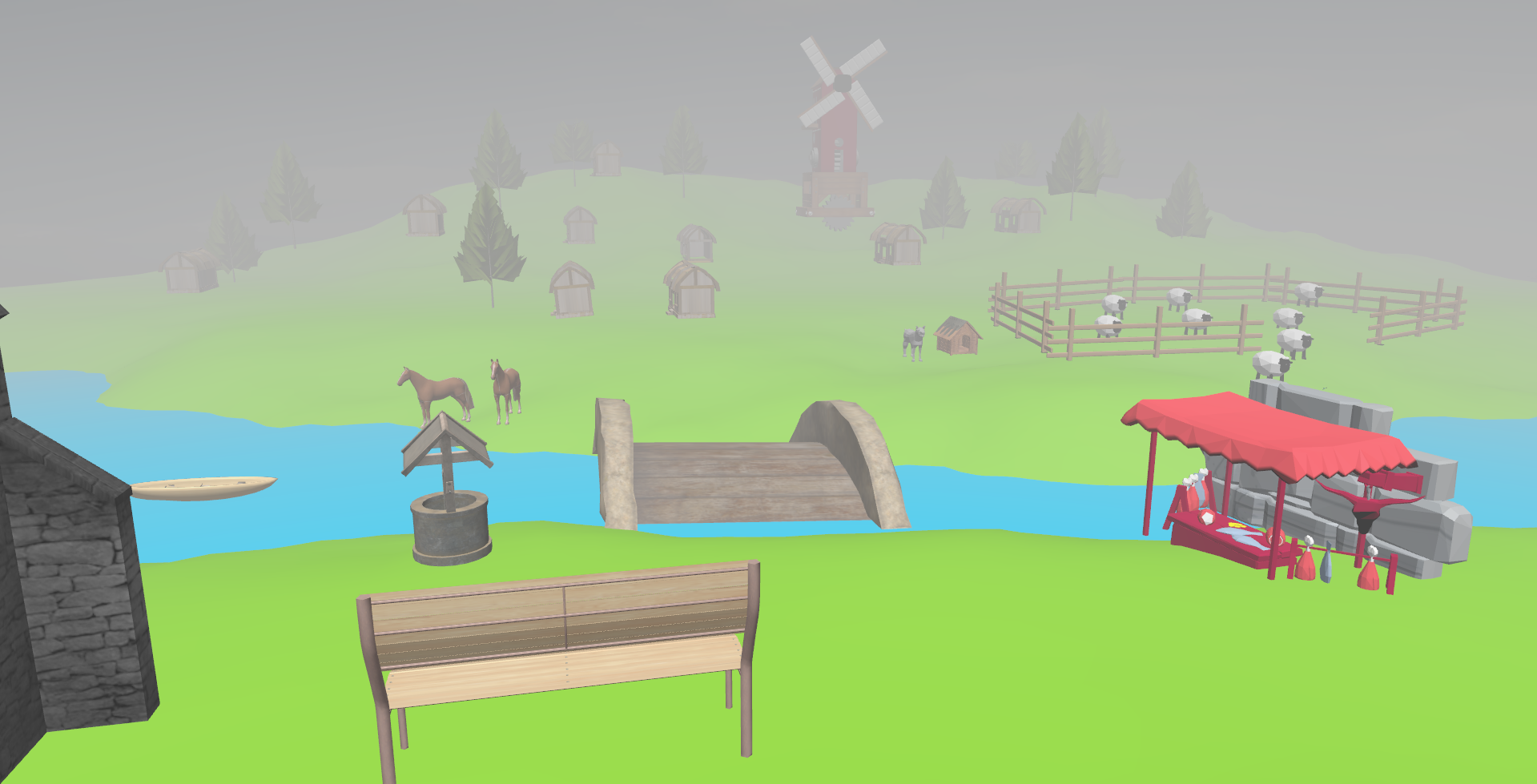


### 

### 2.2 Funcționalitati

Pentru vizualizarea scenei se vor folosi tastele W,A,S,D,Q,E, dar si mouse-ul in caz ca dorim. Apasand tastele Z,X,C utilizatorul va putea modifica vizualizarea scenei in modurile:  
1) Wireframe mode – Z  
2) Solid mode – X  
3) Poligonal mode – C

De asemenea pentru o scurta prezentare a scenei acompaniata de un soundtrack pe fundal, utilizatorul poate apasa tasta P care va face un mic overview al scenei. Pentru oprirea soundtrack-ului se va apasa tasta R. Scena mai prezinta si functionalitatea de ceata, tehnica preluata din laboratoare:



Scena mai contine animatia unei barci aflate in rau:





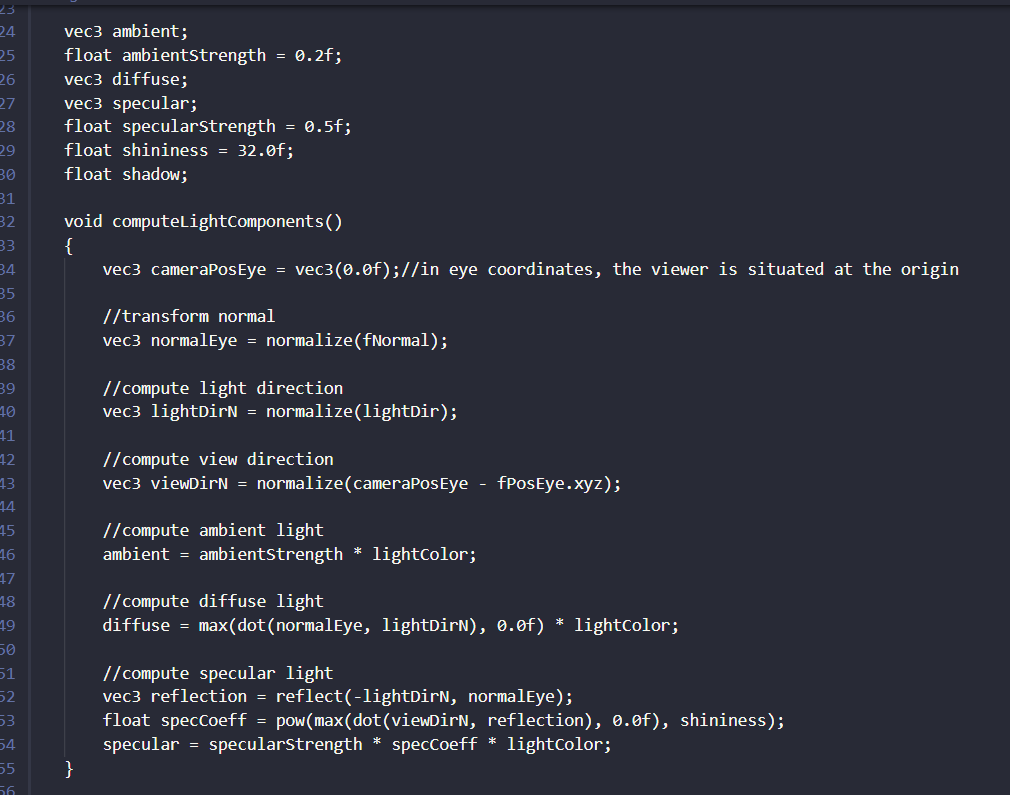
Texturarea mai buna a scenei folosind tehnicile invatate in laborator pentru computeLightComponents:   

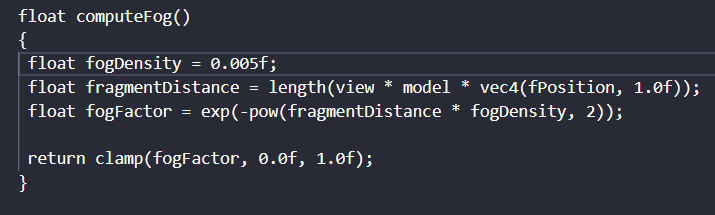

# 3. Implementare

### 3.1 Funcții și algoritmi

Am implementat clasa Camera care este definită pentru a gestiona vizualizarea scenei 3D. Constructorul inițializează poziția, ținta și direcția verticală a camerei, precum și calculează direcțiile frontale și laterale. Metoda **getViewMatrix** utilizează funcția **glm::lookAt** pentru a returna matricea de vizualizare, care reprezintă transformarea vizualizării în spațiul camerei.

In fragment shader si vertex shader am utilizat tehnicile pentru ceata (compute fog) si am incercat de asemenea computeLightComponents:



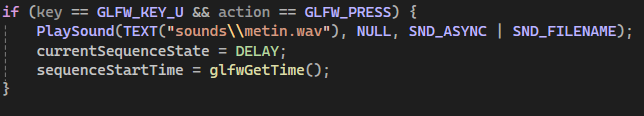


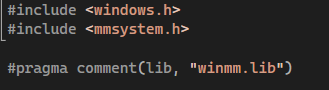
:

In main.cpp, mai exact in processMovement, pe langa atribuirea butoanelor care ajuta la deplasarea camerei, am creat o functie de prezentare sub forma de animatie a scenei folosind tasta U. Aceasta are rolul de a face un mic overview asupra scenei cu acest SequenceState care ma ajuta sa fac mai multe operatii de miscare a camerei folosind un singur buton:



Pentru o ambianta mai placuta, am adaugat librariile <windows.h> si <mmsystem.h> pentru a adauga muzica de fundal scenei mele:





Pentru animatia barcii care se deplaseaza dintr o pozitie in alta si apoi revine la pozitia initiala am folosit un nou model pe care l-am introdus ulterior in programul OpenGL:



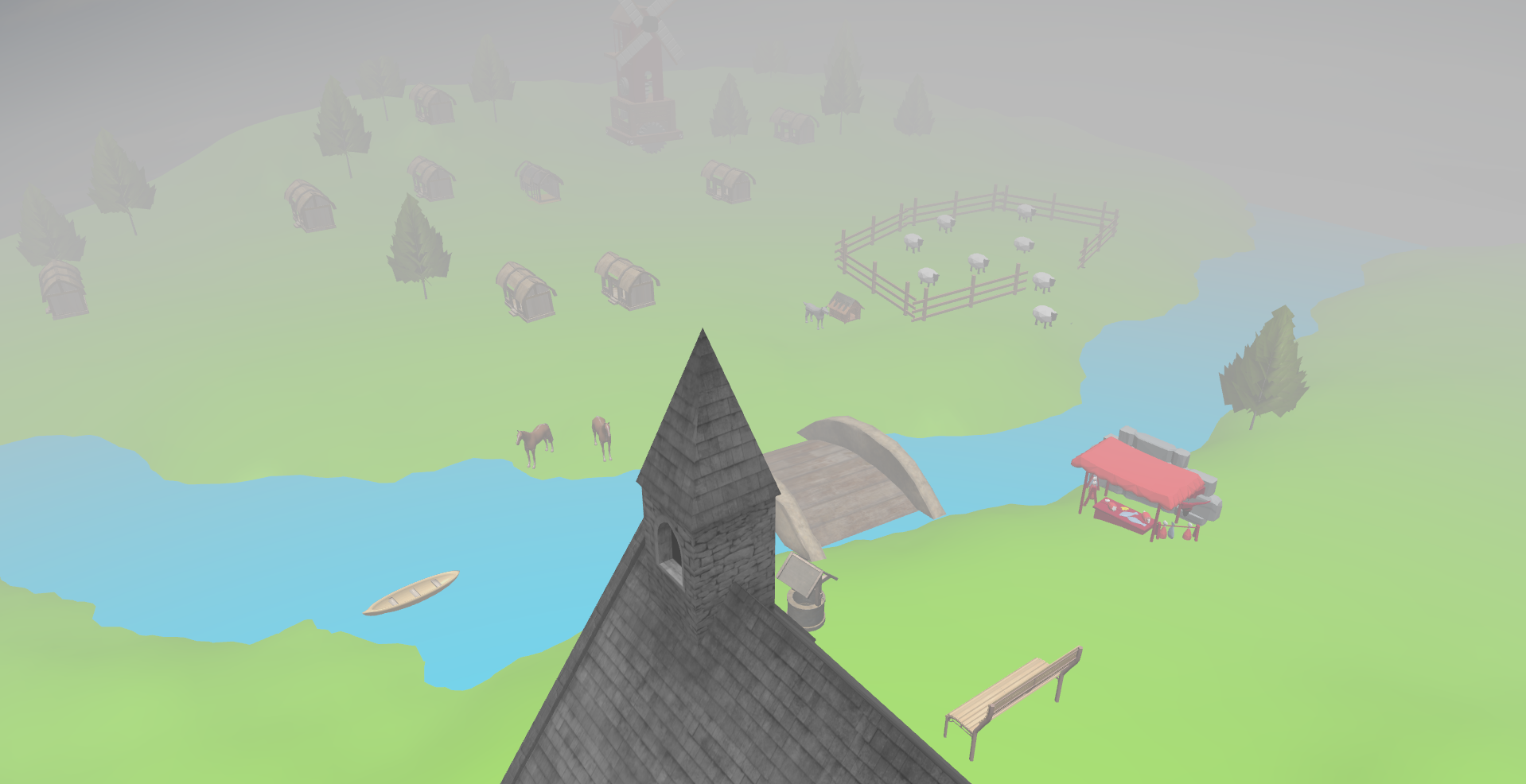
Funcția **initUniforms()** este esențială în configurarea programului nostru, deoarece în această etapă se atribuie și se inițializează locațiile variabilelor uniforme din shader-e, stabilind valori precum detaliile iluminării, perspectiva camerei și matricea normalilor. Această funcție este vitală pentru funcționarea corectă a aplicației. De asemenea, doresc să subliniez importanța funcțiilor **initModels()**, care ne permite să încărcăm modele 3D în OpenGL, și **initShaders()**, unde se încarcă și se compilează shader-ele. În mod obișnuit, fiecare obiect este redat folosind un shader specific.

Pentru imbunatatirea interactiunii cu scena au mai fost adaugate functiile **mouseCallback()** și **processMovement()**.

Am reusit sa implementez si partea functionala de a vedea in wireframe, solid mode si mod polygon cu puncte

### 3.2. Modelul grafic

Scena de sus cu ceata:



Scena in modul wireframe:



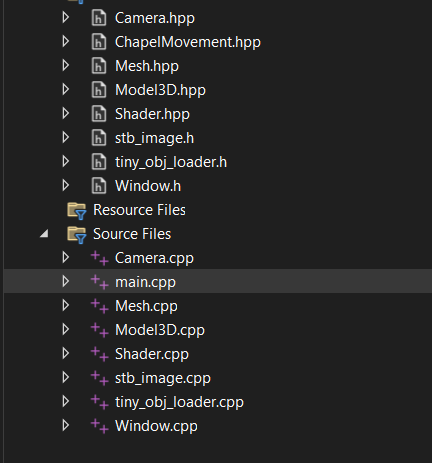
Scena in modul poligonal cu puncte:

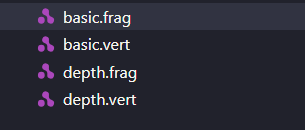


3.3.Structuri de date

În cadrul proiectului, am integrat structuri de date standard pentru lucrul cu OpenGL, incluzând **vec3**, **vec4**, **mat3**, **mat4**, care sunt furnizate de biblioteca GLM pentru manipularea vectorilor și matricelor. În plus, am utilizat instanțe ale claselor personalizate, precum **Camera**, **Model3D**, și **Shader**, pentru a gestiona diverse aspecte ale scenei 3D. De asemenea, s-au folosit tipuri de date GLM, cum ar fi **GLuint** și **GLfloat**, datorită compatibilității lor îmbunătățite cu funcțiile OpenGL, asigurând astfel o integrare eficientă și performanță optimă în procesul de rendering.

# 3.4. Ierarhia de clase





# 5. Interfața utilizator

W - deplasare stânga

A – deplasare dreapta

S – deplasare fața

D – deplasare spate

Q – plasarea view-ului in sus

E - plasarea view-ului in jos  
 Mouse- rotire 360, schimbare unghi de vizualizare

Z – wireframe mode

X – solid mode

C – polygonal mode

R – stop background music

P – start background music

U – prezentarea scenei + start background music

# 6. Concluzii si dezvoltari ulterioare

Consider ca acest proiect a fost unul foarte benefic pentru dezvoltarea skill-urilor in Blender, OpenGL si nu numai. A fost un proiect care a reusit sa ma ajute sa ma folosesc cat mai mult de research si sa imbin tot ce am la dispozitie.  
Ca si dezvoltari ulterioare as vrea sa mai adaug animatii pentru deplasarea personajelor care sa faca atat miscarea picioarelor cat si a corpului. As mai dori sa adaug efecte de ploaie si vant si sa imbunatatesc pe cat posibil scena

# 7. Referințe

* Lucrarile de laborator
* <https://free3d.com/>
* <https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM>
* <https://sketchfab.com/3d-models/old-bible-f0ed46d723dd441bb592d715db241097>
* <https://www.turbosquid.com/>
* Tutorialele blender