

# DOCUMENTAȚIE PROIECT PRELUCRARE GRAFICĂ

Nume: Gafița Diana

Grupa: 302327

An: 3

# Cuprins

1. Prezentarea temei
2. Scenariul
  - 2.1. Descrierea scenei și a obiectelor
  - 2.2. Funcționalități
3. Detalii de implementare
  - 3.1. Funcții și algoritmi
    - 3.1.1. Soluții posibile
    - 3.1.2. Motivarea abordării alese
  - 3.2. Modelul grafic
  - 3.3. Structuri de date
  - 3.4. Ierarhia de clase
4. Prezentarea interfeței grafice utilizator / manual de utilizare
5. Concluzii și dezvoltări ulterioare
6. Referințe

## 1. Prezentarea temei

Subiectul acestui proiect este prezentarea obiectelor 3D folosind biblioteca OpenGL. OpenGL este o interfață de programare a aplicațiilor (API) folosită pentru redarea graficelor vectoriale 2D și 3D. API-ul este de obicei folosit pentru a interacționa cu o unitate de procesare grafică (GPU), pentru a obține randarea accelerată hardware. Am ales ca scena de implementare imaginea unei bucațele de Egipt, integrând astfel simboluri specifice. Scena obiectului se poate manipula cu ajutorul mouse-ului și al tastaturii, utilizatorul având posibilitatea de deplasare prin întreaga scenă.

## 2. Scenariul

### 2.1. Descrierea scenei și a obiectelor

Scena reprezintă o vizualizare minimalistică a Egiptului. Am integrat mai multe simboluri specifice:

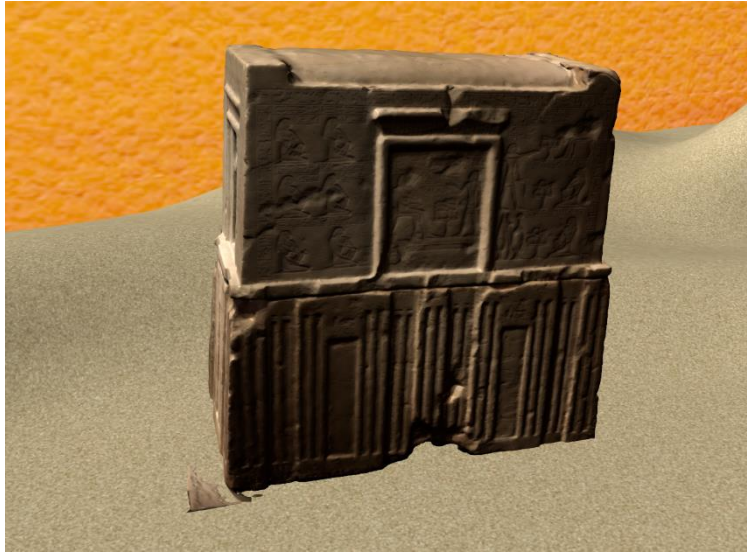
- Piramide



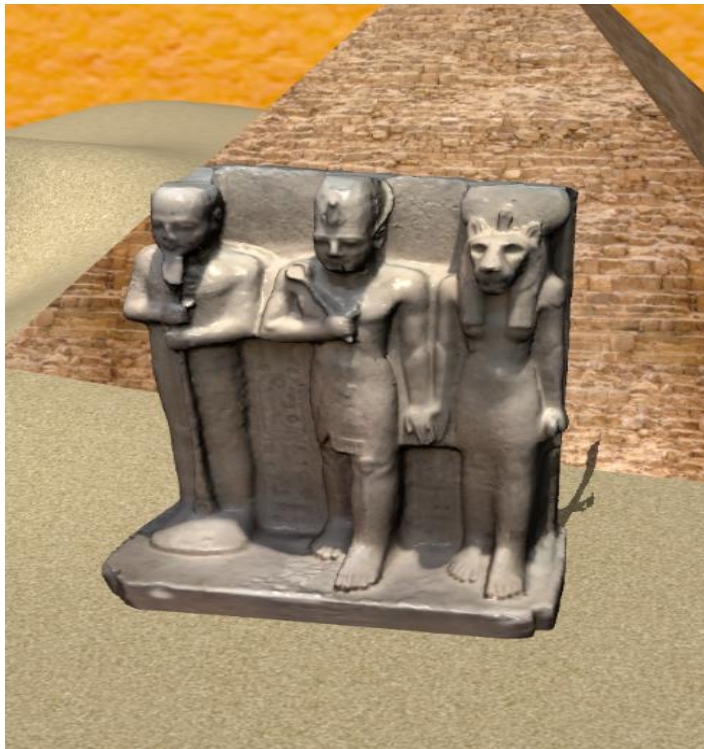
- Statuie îngropată în nisip



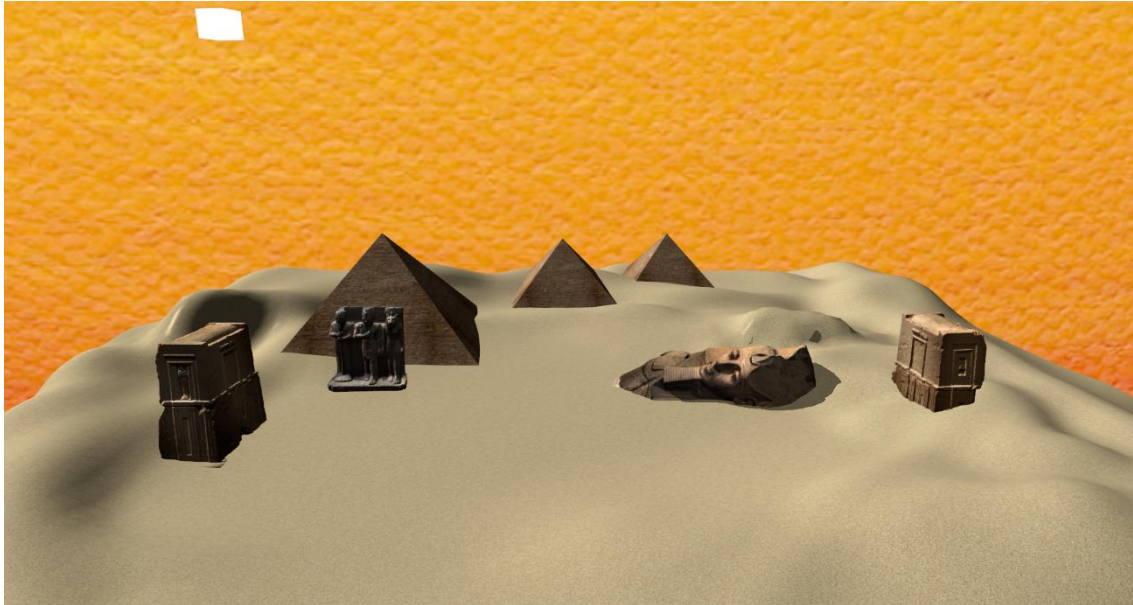
- Coloană gravată cu simboluri egiptene



- Statuie a unor faraoni



Scena este prezentată în deșert, totul fiind înconjurat de nisip.



## 2.2. Funcționalități

Utilizatorul poate vizualiza întreaga scenă prin apăsarea butoanelor de la tastatură sau a mouse-ului. Se poate comuta, de asemenea, între vizualizarea solidă, wireframe sau poligonală folosind butoanele de la tastatură. Există o sursă de lumină: lumina direcțională, care își poate modifica poziția de la tastatură. Utilizatorul are la dispoziție prezentarea scenei, tot din comanda de la tastatură.

Umbrele: pentru a le crea avem nevoie de un depth map care o să conțină valori de adâncime, iar umbra o să se aplice în momentul în care z-ul obiectului e mai mare decât z-ul de pe mapă.

## 3. Detalii de implementare

### 3.1. Funcții și algoritmi

#### 3.1.1. Soluții posibile

Biblioteca OpenGL ne pune la dispoziție o multitudine de funcții, cum ar fi:

- `glfwCreateWindow(...)` folosit pentru crearea ferestrei
- `glGenTextures(...)`
- `glBindTexture(...)`

Una dintre funcțiile folosite în proiect este `void renderScene()`, folosită pentru a trimite toate datele către shaders și a calcula toate valorile. Aici este apelată funcția `drawObjects()` în care sunt toate modelele, se calculează matricea normală, se realizează rotația, translația și

scalarea. O altă funcție este void initUniforms(), unde se afla detaliile luminilor - lumina direțională și cea punctiformă. Funcția void initShaders() este funcția în care se instanțiază shader-urile. Funcția void initModels() este funcția în care sunt instanțiate toate modelele 3D. Alte funcții importante sunt:

- void processMovement()
- void mouseCallback ()
- void keyboardCallback ()

Aceste funcții efectuează toate modificările necesare pe modele sau pe cameră.

Funcția mouseCallBack este implementată prin utilizarea unghiurilor Euler. O bibliotecă de matematică adaptată pentru OpenGL.

### 3.1.2. Motivarea abordării alese

După studierea laboratoarelor în care ni s-au dat exemple de aplicare a diferitelor funcții și pornind de la modelul de bază prezentat acolo, am ales să implementez o scenă simplă folosind noțiunile învățate: calculul luminii, a umbrelor, operații de translație, scalare și rotație.

## 3.2. Modelul grafic

Obiectele și texturile au fost luate de pe internet. Majoritatea obiectelor au fost importate în Blender, editate și apoi plasate în scena finală. Găsirea obiectelor potrivite și plasarea acestora în scenă a fost interesantă dar a și necesitat muncă; a ajutat faptul că sunt destul de multe variante de obiecte pe internet.

## 3.3. Structuri de date

Structurile de date necesare au fost destul de simple. Majoritatea au fost deja implementate în scheletul proiectului pe care am lucrat în timpul laboratorului. Am adăugat în plus unele funcții pentru a calcula lumina.

## 3.4. Ierarhia de clase

- Camera: conține implementarea pentru mișcarea camerei (sus, jos, față, spate, stânga, dreapta)
- SkyBox: conține implementarea skybox-ului (initializare, desenare, încărcarea texturilor)
- Mesh: reprezintă un obiect 3D; include următoarele structuri de date:

Vertex (poziția, vectorul normal și coordonatele texturilor)

Texture (id, tip și cale);

Material

- Model3D: conține metode pentru imprimarea rețelelor folosind un program de shader specificat
- Shader: conține metode de creare și activare a shaderelor

#### 4. Prezentarea interfeței grafice utilizator / manual de utilizare

Utilizatorul poate interacționa cu scena folosind următoarele butoane de pe tastatură:

- 1 – line view
- 2 – point view
- 3 – solid view
- W – miscare în fata în scena
- S – miscare în spate în scena
- A – miscare la stanga în scena
- D – miscare la dreapta în scena
- Q – rotire spre dreapta a scenei
- E – rotire spre stanga în scena
- P – start preview
- O – stop preview
- N – miscare spre dreapta a lightcube-ului
- M – miscare spre stanga a lightcube-ului

#### 5. Concluzii și dezvoltări ulterioare

Lucratul la acest proiect a fost interesant și deși nu cunoșteam aplicația Blender, s-a dovedit a fi foarte creativă. După ce am studiat mai multe tutoriale, a fost destul de ușor să înțeleg majoritatea funcționalităților ca mai apoi să modelez și să-mi creez propria scenă. La partea de VisualStudio am întâmpinat ceva dificultăți în folosirea funcțiilor oferite de OpenGL.

Proiectului i s-ar putea aduce multe îmbunătățiri, de la aranjarea obiectelor în scenă și până la textura acestora, scena ar putea fi mai dinamică și mai complexă.

#### 6. Referințe

- [https://www.youtube.com/watch?v=m34EamE5biQ&list=PLrgcDEgRZ\\_kndoWmRkAK4Y7ToJdOf-OSM&index=5&ab\\_channel=CosminNandra](https://www.youtube.com/watch?v=m34EamE5biQ&list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM&index=5&ab_channel=CosminNandra)
- [https://sketchfab.com/search?sort\\_by=-likeCount&type=models](https://sketchfab.com/search?sort_by=-likeCount&type=models)
- [How to make an Atmospheric Scene under 10 mins! | Blender 2.8 Tutorial - YouTube](#)