Documentație proiect

Prelucrare grafică

Buruian Cătălina

Grupa 30231

1.Cuprins

**1. Cuprins1**

**2. Prezentarea temei2**

**3. Scenariul3**

3.1. Descrierea scenei și a obiectelor3

3.2. Funcționalități3

**4. Detalii de implementare4**

4.1. Funcții și algoritmi4

4.1.1. Soluții posibile4

4.1.2. Motivația abordării alese4

4.2. Modelul grafic4

4.3. Structuri de date4

4.4. Ierarhia de clase4

**5. Manual de utilizare5**

**6. Concluzii și dezvoltări ulterioare6**

**7. Referințe7**

# 2. Prezentarea temei

Proiectul în cauză a presupus realizarea unei scene de obiecte 3D într-un mod cât mai realist și estetic, utilizând librăriile OpenGL, GLFW și GLM. Pentru a plasa obiectele în scenă și a mapa texturi pe acestea se folosește aplicația Blender, care pune la dispoziția utilizatorului o multitudine de funcționalități. Obiectele 3D modelate în Blender atât cele individuale, cât și ansamblul lor (scena de obiecte) vor fi exportate ca fișiere .obj în fișierul proiectului, împreună cu texturile acestora (în general fișiere .png).

Utilizatorul se poate mișca în scenă cu ajutorul diferitelor taste, putând vizualiza scena în mai multe moduri (wireframe, solid, smooth și polygonal) și în contextual de zi și noapte.

# 3. Scenariul

## 3.1. Descrierea scenei și a obiectelor

În cadrul acestui proiect am ales să modelez planeta Marte într-un context post-apocaliptic. De aceea, obiectele incluse în scenă sunt o cafenea abandonată, o stație de alimentare, un turn de apă, o casă părăsită, o mașină avariată, stâlpi de iluminat, etc. Pentru a simula ideea de planetă Marte am adăugat și mai multe obiecte de tip roci, vulcani și gheață. Pentru a adăuga puțină viață scenei am introdus și un obiect de tip navă spațială care se mișcă.

## 3.2. Funcționalități

Scena modelată este una care descrie pustietatea pe timp de zi, iar pe timp de noapte devine un mediu obscur acaparat de ceață. Dinamismul scenei este dat doar de OZN-u care zboară în sus și în jos. Pentru a crea detalii cât mai realiste am implementat transparența pentru gheață, sticlă spartă și sticlele de pe terasa cafenelei. Ceața, de asemenea, joacă un rol important în intensificarea atmosferei generale. Sursele de lumină din scenă sunt în număr de cinci: lumina principală(direcțională), două lumini galbene care provin de la stâlpii de iluminat și două lumini verzi care sunt poziționate în zona OZN-ului. Detaliile de implementare vor fi discutate în capitolul următor.

# 4. detalii de implementare

Implementarea proiectului a constat dintr-un continuu proces de învățare și aprofundare a cunoștiințelor deja dobândite în cadrul laboratorului de prelucrare grafică.

## 4.1. Funcții și algoritmi

Algoritmii implementați în cadrul proiectului sunt următorii: generarea ceței, aplicarea transparenței pe obiecte, crearea de lumini de tip direcțională, punctiformă și spotlight.

### 4.1.1 soluții posibile

Pentru generarea ceței, am implementat ceața exponențială pătratică care este o îmbunătățire a ceței exponențiale, ținând cont de reducerea intensității luminii în funcție de distanță. Factorul de atenuare care va fi folosit reprezintă densitatea de ceață, această densitate fiind constantă în toată scena. Rezultatul este o scădere rapidă a factorului de ceață în comparație cu alte abordări. Efectul realist este dat și de setarea culorii pentru ceață astfel încât să se potrivească cu skyBox-ul și culorile scenei.

Pentru aplicarea transparenței pe anumite obiecte, cum ar fi pe gheață, sticle și cioburile de sticlă, am utilizat tehnica de blending din OpenGL, apelând funcția glEnable(GL\_BLEND). Cu ajutorul variabilei uniform “alpha” am trimis gradul de transparență al obiectelor.

Pentru implementarea mai multor tipuri de lumini am folosit funcții diferite care calculează seturi de lumini ambientală, difuză, speculară. Lumina direcțională se calculează după modelul de iluminare Blinn-Phong, iar în calculul luminilor punctiforme și spotlights se ține cont de direcția luminii, factorul de atenuare, respectiv direcția luminii, direcția punctului în care se îndreaptă lumina, unghiul dintre acestea(theta) și unghiul care cuprinde lumina(phi).

### 4.1.2 Motivația abordării alese

Majoritatea modurilor de implementare au fost alese pentru a aduce un plus de realism scenei și pentru a contribui la estetica acesteia. Funcționalitățile au la bază abordări simpliste, care au necesitat în principal înțelegerea bibliotecilor și a funcțiilor din OpenGL.

## 4.1. modelul grafic

Modelul grafic al acestui proiect a fost realizat folosind modelul de iluminare Blinn-Phong, deoarece în plus față de modelul original, îmbunătățește performanța, evitând calculul costisitor al vectorului de reflexie. În procesul de rasterizaream folosit efecte create de lumini, ceață și transparență. Animația OZN-ului a fost realizată printr-o translație pe axa Oy și o rotație completă în jurul originii obiectului. Obiectul se mișcă constant sus-jos, poziția acestuia pe axa Oy fiind încadrată într-un interval și incrementul, respectiv decrementul poziției făcându-se cu un step.

## 4.2. structuri de date

Structurile de date pe care le-am folosit în acest proiect sunt structurile specifice OpenGL: VAO, VBO, EBO și FBO. VAO (Vertex Array Object) este un obiect care conține unul sau mai multe VBO-uri și este creat pentru a reține informații necesare rasterizării obiectelor. VBO (Vertex Buffer Object) este un buffer de memorie care descrie coordonatele sau culorile vârfurilor. EBO (Element Buffer Array) stochează indicii pentru evitarea duplicatelor. Indicii vor fi utilizați de OpenGL pentru a decide ce vârfuri să aleagă atunci când formează obiectele. FBO (Frame Buffer Object) este o colecție de texture sau alte tipuri atașate. Pe lângă structurile de date specifice OpenGL, am folosit și structuri de date din biblioteca glm, cum ar fi vec3, vec4, mat3, mat4 etc.

## 4.2. Ierarhia de clase

Clasa „Camera” definește parametrii camerei și mișcările acesteia.

Clasa „Model3D” este destinată prelucrării obiectelor 3D din scenă, conținând funcții de aplicare a texturii pe obiect, în funcție de fișierul .mtl și funcții de încărcare a obiectului în scenă.

Clasa „Skybox” definește pozițiile vârfurilor texturii ce urmează să fie aplicată pe cubeMap.

Clasa „Shader” definește funcționalitatea shaderelor acestui proiect.

Cea mai semnificativă clasă din proiect este clasa „Main” care caracterizează întreaga funcționalitate a proiectului, folosindu-se de celelalte clase.

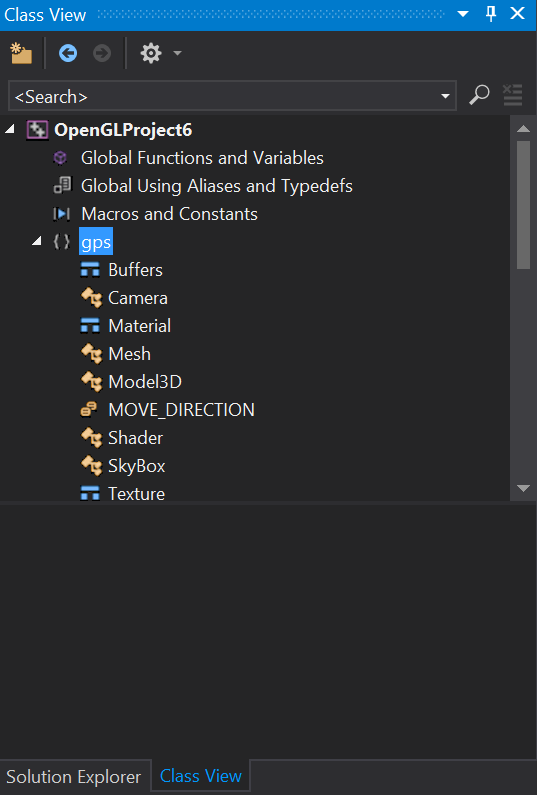


Fig 1. Class View în Visual Studio

# 5. Manual de utilizare

Pentru a rula proiectul, este recomandat ca acesta să fie deschis într-un editor de cod , precum Visual Studio și alegerea modului Debug sau Release. După executarea programului, se va deschide o fereastră cu scena, iar pentru a naviga prin scenă sau a vizualiza scena în diferite moduri, următoarele taste sunt disponibile:

Tasta Q – mod de prezentare

Tasta H – modul de zi

Tasta J – modul de noapte

Tasta W – mișcare cameră înainte

Tasta S – mișcare cameră înapoi

Tasta săgeata stânga – mișcare cameră la stânga

Tasta săgeata dreapta – mișcare cameră la dreapta

Tasta săgeata în sus – mișcare cameră în sus

Tasta săgeata în jos – mișcare cameră în jos

Tasta R – rotire cameră la dreapta

Tasta T – rotire cameră la stânga

Tasta P – vizualizare mod wireframe

Tasta O – vizualizare mod solid

Tasta U – vizualizare mod smooth

Tasta Y – vizualizare mod polygonal

Figurile de mai jos sunt reprezentative pentru modul în care arată scena.



Fig 2. Cadru cafenea pe timp de zi

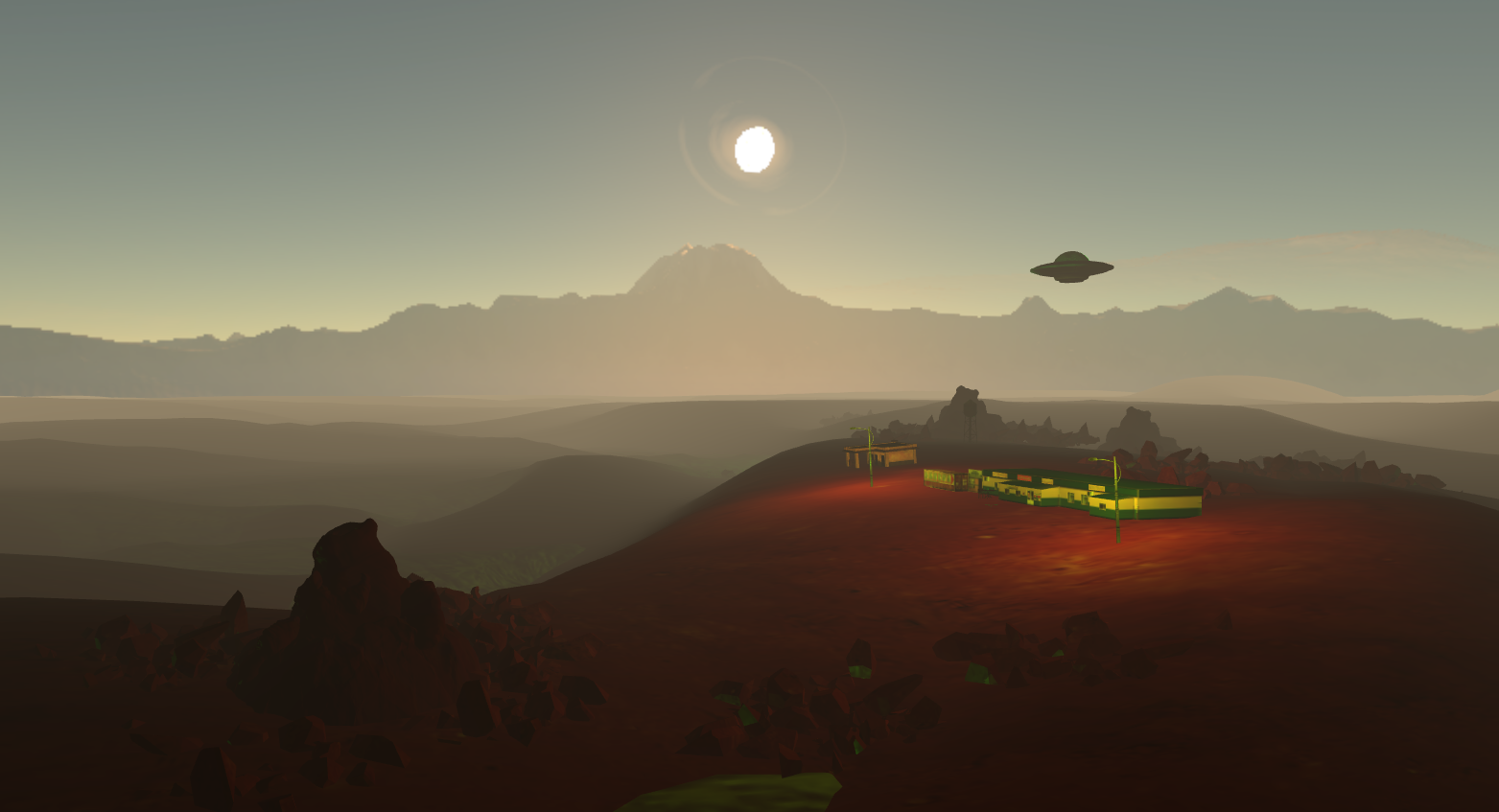


Fig 3. Cadru scena pe timp de noapte



Fig 4. Cadru OZN și lumini verzi

# 6. concluzii și dezvoltări ulterioare

Ca și dezvoltări ulterioare, se mai pot adăuga obiecte care să se potrivească cu tema scenei, se mai pot realiza efecte de reflecție sau umbre. De asemenea, se mai pot adăuga animații mai complexe pentru ca scena să fie mai interactivă.

În concluzie, mi-a plăcut acest proiect și modul în care arată scena finală. Mi s-a părut destul de interesant în special să manipulez obiecte 3D cu ajutorul librăriei OpenGL.

# 7. referințe

[1] Laboratorul de prelucrare grafică

[2] <https://learnopengl.com>