Prelucrare Grafică - OpenGL

DOCUMENTAȚIE

Student: Tecuci Dragoș

Grupa: 30232

Cuprins

1. Prezentare temei 2

2. Scena 2

2.1 Descriere scena si obiecte 3

2.2 Funcționalități 4

3. Detalii de implementare 4

3.1 Funcții si algoritmi 4

3.2 Modelul grafic 5

3.3 Structuri de date 5

3.4 Ierarhia de clase 6

4 Prezentarea interfeței grafice utilizator / manual de utilizare 7

5 Concluzii si dezvoltări ulterioare 8

6 Referințe 9

1. Prezentarea temei

Tema proiectului se axează pe dezvoltarea unei aplicații interactive ce combină aspecte ale graficii 3D, animației și interacțiunii utilizatorului.

Principalele componente ale temei includ:

1. **Grafică 3D cu OpenGL (GLFW/GLM) :** Utilizarea bibliotecii OpenGL pentru crearea și afișarea unui mediu 3D. Aceasta include gestionarea obiectelor 3D, texturarea, iluminarea și crearea unui peisaj urban realist.
2. **Sistem de Cameră Interactivă:** Implementarea unei camere de tip "first-person" pentru a oferi utilizatorului controlul asupra explorării scenei. Cameră permite mișcări continue în spațiul 3D.
3. **Animarea Camerei:** Integrarea unor animații fluide pentru mișcarea și rotirea camerei. Aceste animații includ tranziții între poziții diferite, adăugând un aspect cinematic experienței.
4. **Interacțiunea Utilizatorului:** Permiterea utilizatorului să interacționeze cu mediul virtual prin tastatură sau alte dispozitive de intrare. Acest aspect adaugă un nivel suplimentar de imersiune în scenă.
5. **Documentație și Explicare:** Crearea unei documentații cuprinzătoare pentru a explica funcționalitățile, structura și utilizarea proiectului. Aceasta include descrierea codului sursă și explicații pentru eventuale aspecte tehnice.

2. Scenariul

2.1. Descriere scenă și obiecte

Scena se desfășoară într-un vast deșert, un loc uitat de timp și populat doar de nisipul arid și pustiu. În mijlocul acestei vastități se înalță un turn vechi, cu pereții săi îmbătrâniți de soare și vânt. În jurul turnului, un râu liniștit curge într-o ambianță de uitare, conturându-l ca o graniță misterioasă între pustietate și oportunitatea descoperirii.

Podul, construit din lemn, traversează râul, oferind un drum către necunoscut. Însă, în proximitatea podului, rămășițe umane sub formă de cranii își arată prezența, sugerând că nu orice călător a avut parte de o călătorie lină prin acest loc abandonat.

În apropiere de pod, un automobil ruginit își găsește refugiul, ca un monument al trecutului și al încercărilor îndeplinite sau abandonate. Un explorator, pe urmele unui destin necunoscut, a ajuns în acest loc și a descoperit această scenă ireală.

În umbra turnului, se ridică un castel misterios, jumătate îngropat în nisip, ca o amintire a unor vremuri apuse. Fără semne de viață în interior, castelul stă ca o enigmă, așteptând să fie descifrat.

Întregul peisaj emană o atmosferă de abandon și trecere a timpului. Prin unirea elementelor precum turnul, râul, podul, cranii, automobilul și castelul, scenariul dezvăluie o poveste misterioasă, a cărei semnificație rămâne ascunsă în nisipul deșertului, așteptând să fie descoperită de cei curajoși care se aventurează în această țară uitată.

2.2. Funcționalități

Proiectul include elemente de prezentare, animație, evidențiindu-se printr-o tranziție fluidă a camerei între diverse poziții și orientări. Aceste animații contribuie la o experiență vizuală plăcută și dinamică pentru utilizator.

3. Detalii de implementare

3.1. Funcții și algoritmi

Funcția windowResizeCallback() este apelată atunci când fereastra OpenGL este redimensionată și se ocupă de ajustarea matricii de proiecție pentru a menține aspectul corect al scenei.

Funcțiile keyboardCallback() și mouseCallback() gestionează input-ul utilizatorului, inclusiv mișcarea camerei, activarea/dezactivarea luminii, și altele.

Funcția processMovement() este apelată în bucla principală pentru a actualiza starea jocului în funcție de tastele apăsate.

Mai departe, există funcții pentru inițializarea obiectelor, shaderelor, matricilor și framebuffer-ului pentru crearea hărții de adâncime (shadow map). Acestea includ și inițializarea și randarea norilor de ploaie, cu opțiunea de a le activa sau dezactiva.

În bucla principală, funcțiile updateDelta(), renderScene(), renderBoat(), updateBoatPos(), renderRaindrops(), și altele sunt apelate pentru a actualiza și randare diverse aspecte ale scenei. De exemplu, renderScene() se ocupă de randarea scenei principale, inclusiv a umbrelor calculate cu ajutorul hărții de adâncime.

Este important de menționat că acest cod presupune existența unor clase precum `gps::Camera`, `gps::Shader`, `gps::Model3D`, și altele, care sunt definite în sursele corespunzătoare și în fișierele header incluse.

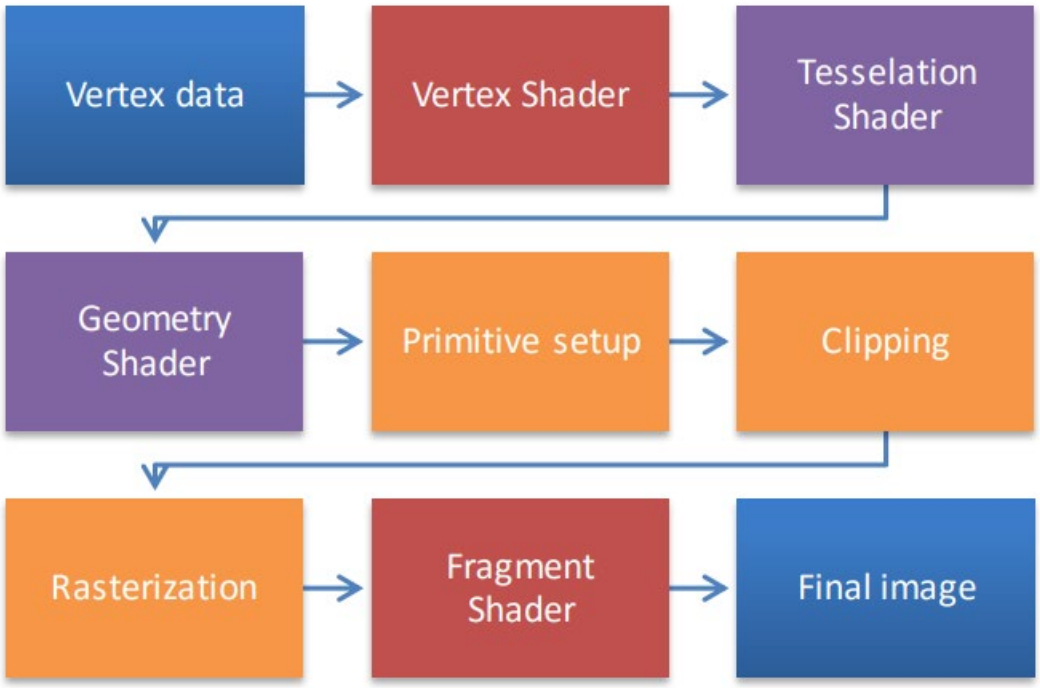
Constructorul camerei este esențial pentru poziționarea și vizualizarea scenei. Prin intermediul acestuia, se poate ajusta unghiul de vizualizare, precum și se pot efectua mișcări sau rotații cu ajutorul tastaturii. Funcția de rotație a camerei folosește informații despre poziția exactă și ultima poziție a mouse-ului.

Implementarea luminii globale se bazează pe modelul Phong, în care se utilizează cele trei componente ale luminii: ambientală, speculară și difuză. Aceste trei tipuri de lumină se combină pentru a ilumina scena, iar logica acestui proces este implementată în shaderul de bază.

Ceața poate fi activată de către utilizator prin intermediul butoanelor. Informația referitoare la densitatea ceții este transmisă shaderului principal, unde se calculează și se amestecă cu culorile și lumina corespunzătoare.

3.2. Modelul grafic

Modelul grafic utilizat în implementare este cel definit de OpenGL:



Nu am programat Shader-ele de Tesselation și Geometry, ci doar Shader-ele de Vertex și Fragment. În Shader-ul de Fragment, am efectuat majoritatea calculelor: pentru lumină direcțională, ceață, umbre, texturi și culori. În Shader-ul de Vertex, am utilizat matricea de transformare a spațiului luminii pentru a calcula poziția curentă a vârfului în spațiul luminii (scena văzută din perspectiva surselor de lumină), astfel încât să pot obține valoarea sa Z (adâncimea) deoarece valorile din harta de adâncime erau în coordonatele luminii și era necesar să fac o comparație pentru a calcula valoarea umbrei.

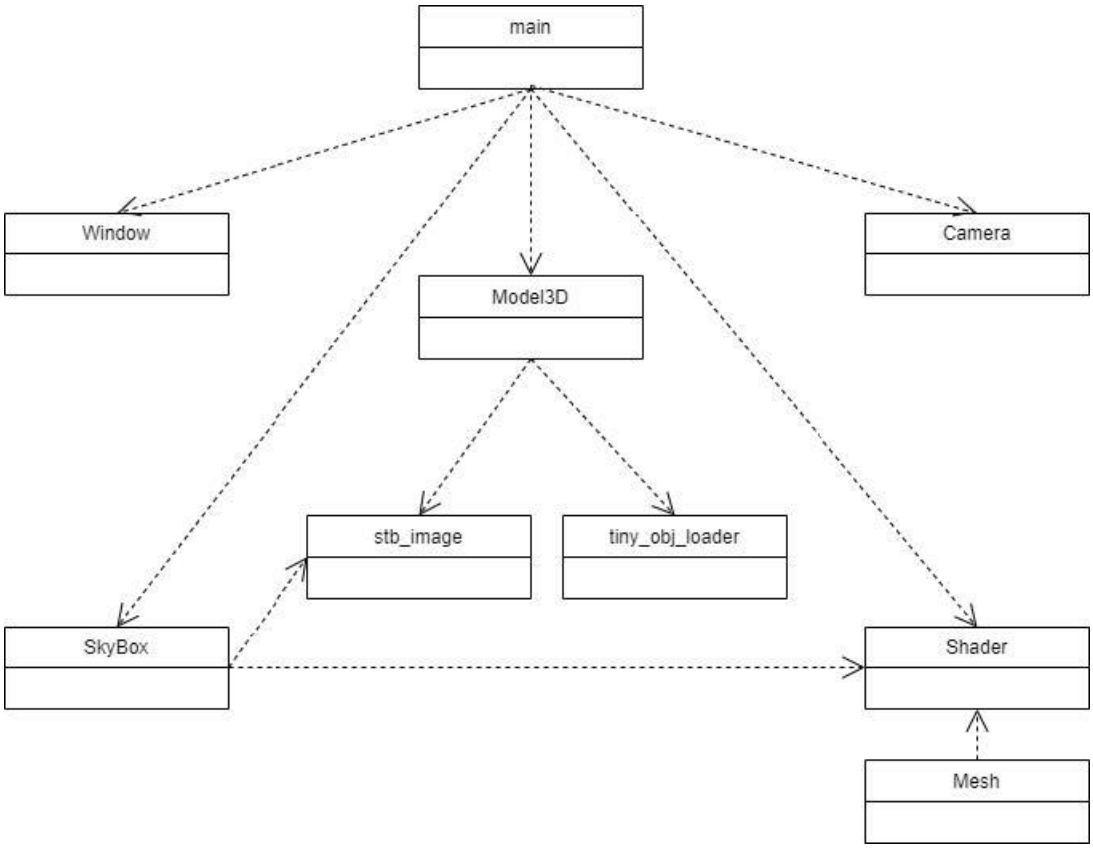
3.3. Structuri de date

Am o structură de date numit `Raindrop` care conține două obiecte `glm::vec3`: `position` și `velocity`. Apoi, o variabilă globală `std::vector<Raindrop> raindrops;` este declarată, reprezentând un vector dinamic de obiecte de tip `Raindrop`.

- `position`: Aceasta stochează poziția tridimensională a picăturii de ploaie în spațiul 3D. `glm::vec3` este un tip de date din biblioteca GLM (OpenGL Mathematics) și reprezintă un vector tridimensional (o tripletă de valori: x, y și z).
- `velocity`: Acesta reprezintă vectorul de viteză al picăturii de ploaie, indicând direcția și magnitudinea (viteza) cu care picătura de ploaie se mișcă în spațiu.
- `std::vector<Raindrop> raindrops;`: Un vector dinamic de obiecte de tip `Raindrop`. Un vector dinamic în C++ (implementat în biblioteca standard prin `std::vector`) este o structură de date care poate crește și scădea în dimensiune în timpul execuției. În acest caz, vectorul conține obiecte de tip `Raindrop`, adică picături de ploaie.

Prin utilizarea acestui vector, programul poate gestiona și manipula multiple picături de ploaie, actualizându-le pozițiile în funcție de vectorii de viteză, adăugând sau eliminând picături de ploaie din scenă.

3.3. Ierarhia de clase



4. Prezentarea interfeței grafice utilizator / manual de utilizare

- Mișcarea camerei:
  - "W" : mișcare în față
  - "A" : mișcare în stângă
  - "S" : mișcare în spate
  - "D" : mișcare în dreapta

- Rotire scenă:
  - "Q" : rotire la stânga
  - "E" : rotire la dreapta

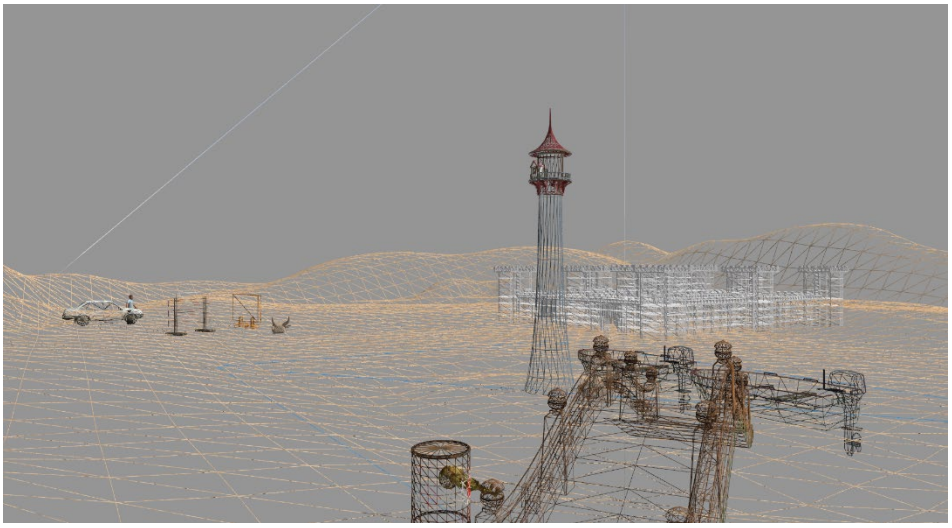
- Manipularea direcției luminii (se vizualizează astfel și mișcarea umbrelor):
  - "U" : rotire la stânga
  - "I" : rotire la dreapta

- Ceață:
  - "F" : pornire efect de ceață
  - "G" : oprire efect de ceață

- Resetarea poziției camerei + animație de prezentare a scenei : "R"

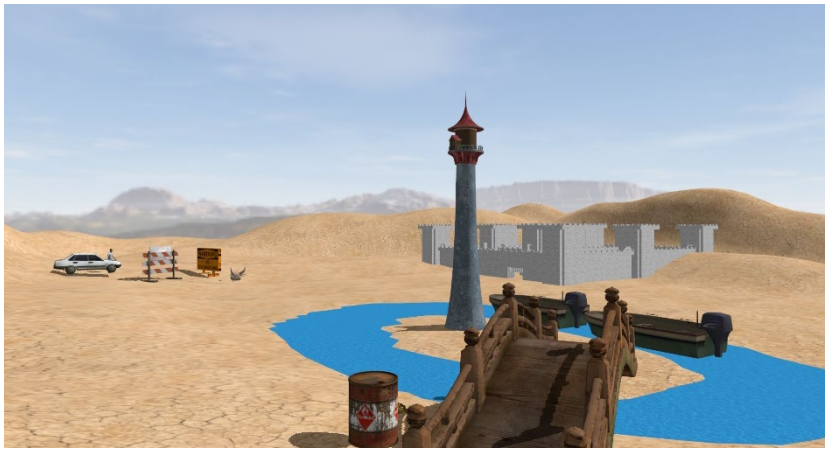
- Rotire cameră : cu ajutorul unui Mouse

- Moduri de vizualizare :
  - "L" : wireframe

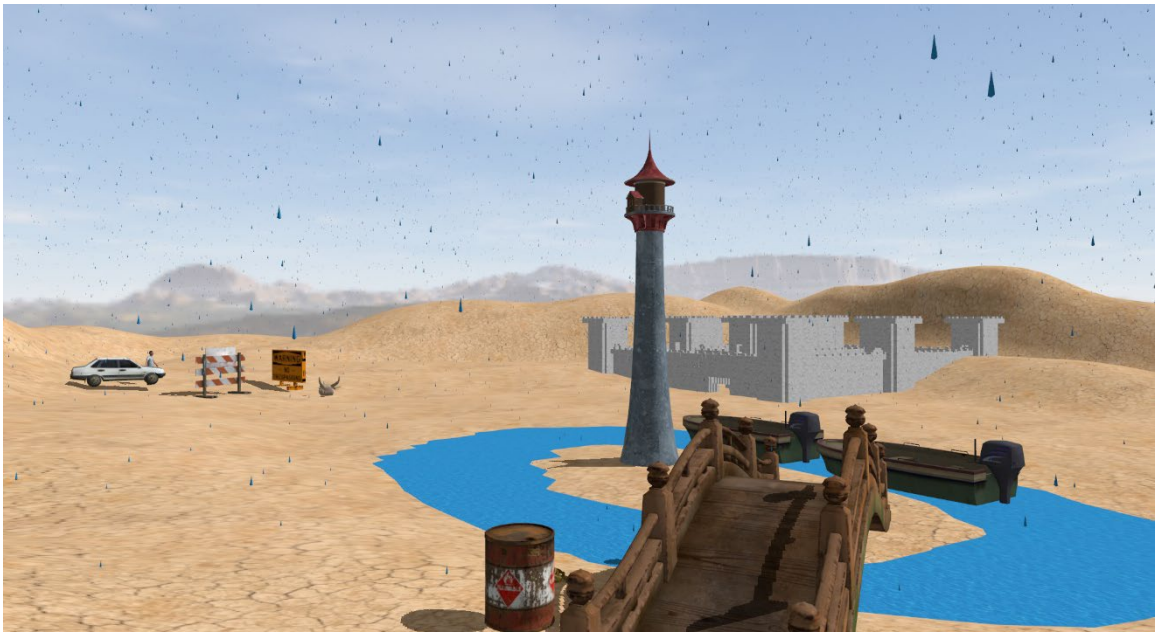


  - "J" : point

- "K" : solid



- ❖ Pornire / oprire ploaie: "H"



- ❖ Direcție de mișcare a unei bărci din scenă (un obiect): ↑, ↓, ←, →
- ❖ Animație barcă : "N"

5. Concluzii și dezvoltări ulterioare

Implementarea acestui proiect a oferit o experiență valoroasă în dezvoltarea unei aplicații grafice folosind OpenGL. Modelul grafic bazat pe OpenGL a permis realizarea unui mediu 3D interactiv, oferind utilizatorului posibilitatea de a interacționa cu diverse elemente precum barca, lumina globală, și efectele atmosferice.

Prin intermediul shaderelor și tehnologiilor OpenGL, am reușit să implementez o lumina globală și un efect de ceață. De asemenea, manipularea camerei și a obiectelor din scenă s-a dovedit a fi fluidă și intuitivă, contribuind la experiența generală a utilizatorului.

Dezvoltări ulterioare:

Proiectul oferă o bază solidă pentru extinderi și îmbunătățiri ulterioare. Câteva direcții pentru dezvoltări ulterioare ar putea include:

1. **Optimizări de performanță:** Îmbunătățirea performanței prin optimizări ale shaderelor sau implementarea tehnicilor avansate de randare poate contribui la o experiență mai fluidă.
2. **Extinderea funcționalității:** Adăugarea de funcționalități suplimentare precum mai multe obiecte interactive, animații sau elemente de interfață ce vor îmbogăți experiența utilizatorului.
3. **Realizarea unui gameplay mai complex:** Introducerea unui gameplay mai complex, misiuni sau obiective pentru jucător ar putea transforma proiectul într-un joc interactiv și captivant.
4. **Optimizări vizuale:** Detaliile vizuale pot fi îmbunătățite prin adăugarea de texturi de înaltă rezoluție, modele 3D mai complexe sau efecte vizuale avansate.

5. Bibliografie

Majoritatea resurselor utilizate sunt cele implementate pe parcursul laboratoarelor de Prelucrare Grafică

- https://learnopengl.com/
- https://www.glfw.org/docs/latest/quick.html
- https://open.gl/
- https://www.youtube.com/playlist?list=PLrgcDEgRZ_kndoWmRkAK4Y7ToJdOf-OSM
- https://www.cgtrader.com/
- https://sketchfab.com/
- http://www.tutorialsforblender3d.com/Skybox/Page1