



**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

Facultatea de Automatică și Calculatoare

Prelucrare Grafică

# Proiect în OpenGL

## -Mushroom World-



Profesor îndrumător:  
Adrian Sabou

Student:  
Dunca Denisa Mihaela  
Grupa: 30237

2021-2022



## **Cuprins**

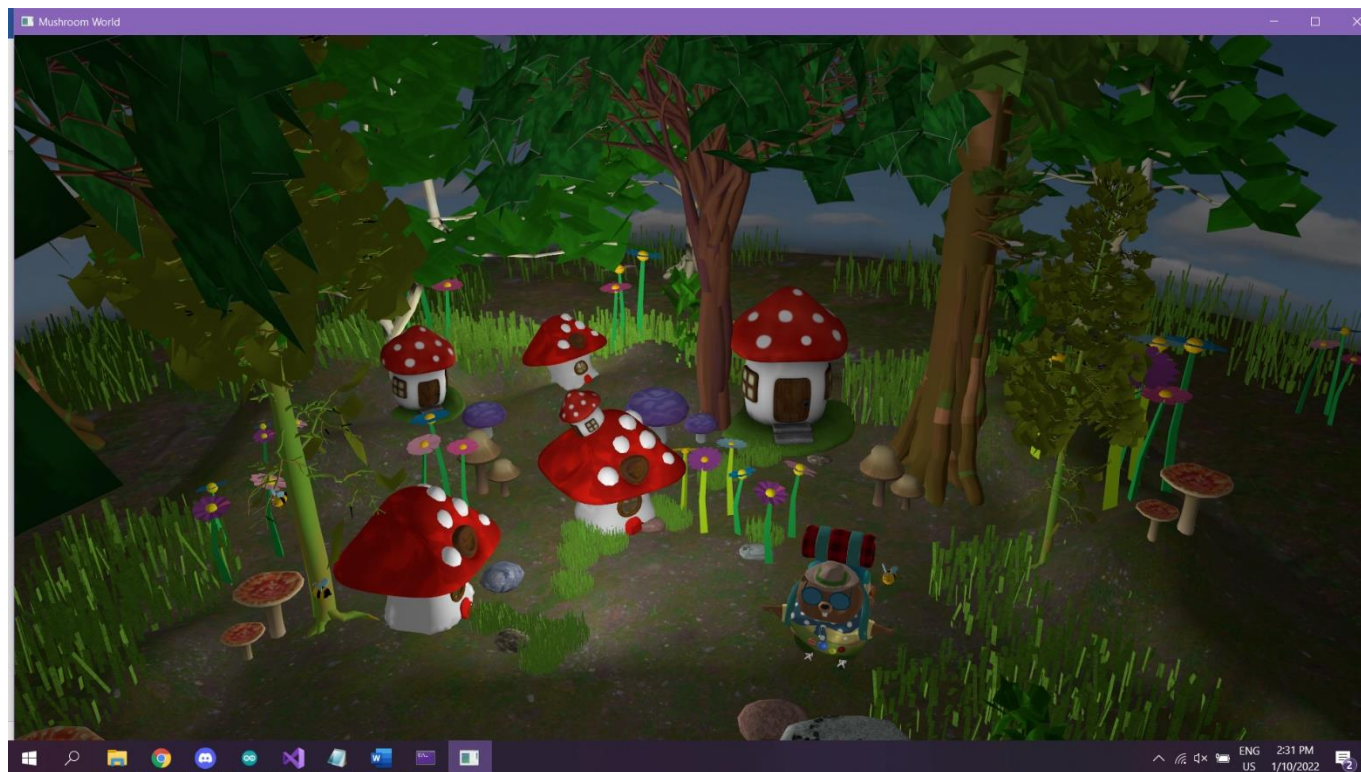
1. Prezentarea temei.....	3
2. Scenariul .....	3
2.1 Descrierea scenei și a obiectelor.....	4
2.2 Funcționalități.....	4
3. Detalii de implementare și algoritmi .....	7
3.1 Algoritmi .....	7
3.1.1 Soluții posibile.....	7
3.1.2 Motivarea abordării alese .....	8
3.2 Modelul grafic .....	8
3.3 Structuri de date.....	8
3.4 Ierarhia de clase .....	8
4. Manual de utilizare .....	8
5. Concluzii și dezvoltări ulterioare.....	9
6. Referințe.....	9

## 1. Prezentarea temei

Proiectul are ca și temă crearea unei scene 3D cu ajutorul bibliotecilor puse la dispoziție de OpenGL. În scena creată trebuie implementate noțiunile discutate și prezentate la laboratorul de Prelucrare Grafică, utilizând ca limbaj de programare C++, iar ca IDE de dezvoltare Visual Studio. Proiectul de față reușește să prezinte un mod de a crea o scenă, de a-i adăuga texturi și două tipuri de lumini (direcțională și de tip point), precum și prezentarea unui mod de creare a animațiilor pe diferite obiecte din scenă.

## 2. Scenariul

Scena 3D creată surprinde o lume a ciupercilor, într-o pădure, înconjurată de flori, copaci, albinuțe și având ca personaj principal un castor cercetaș.



Figură 1 Scena 3D



## 2.1 Descrierea

## scenei și a obiectelor

Scena în sine este alcătuită din trei obiecte principale, un obiect numit "world" care reprezintă întreaga lume, acest obiect este creat în Blender prin importare de diferite obiecte și texturi găsite pe Internet, unele dintre texturi au fost făcute de mine în Paint. Celelalte două obiecte sunt grupul de albinuțe și castorul. Aceste două obiecte au fost adăugate în scenă tot în Blender și exportate separat deoarece acestor două obiecte le-am aplicat diferite animații de mișcare.



Figură 2 Grupul de albinuțe



Figură 3 Castorul cercetaș

## 2.2 Funcționalități

Navigarea în scenă se poate face ușor, utilizând tastele W, A, S, D, Q, E, Left Shift și Space pentru mișcarea camerei și, de asemenea, și prin mișcarea mouse-ului. Tastele 5 și 6 se ocupa de creșterea respectiv scăderea densității ceții, tastele 7 și 8 se ocupă de închiderea respectiv deschiderea nuanței de lumină pentru a crea efectul de noapte și zi în scenă. În plus se pot utiliza tastele 1, 2, 3 și 4 pentru a vedea scena în modurile solid, wireframe, poligonal și smooth. De asemenea se poate activa și dezactiva animațiile albinelor și a castorului utilizând tastele V și M, N.





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA



Figură 4 Scena văzută noaptea

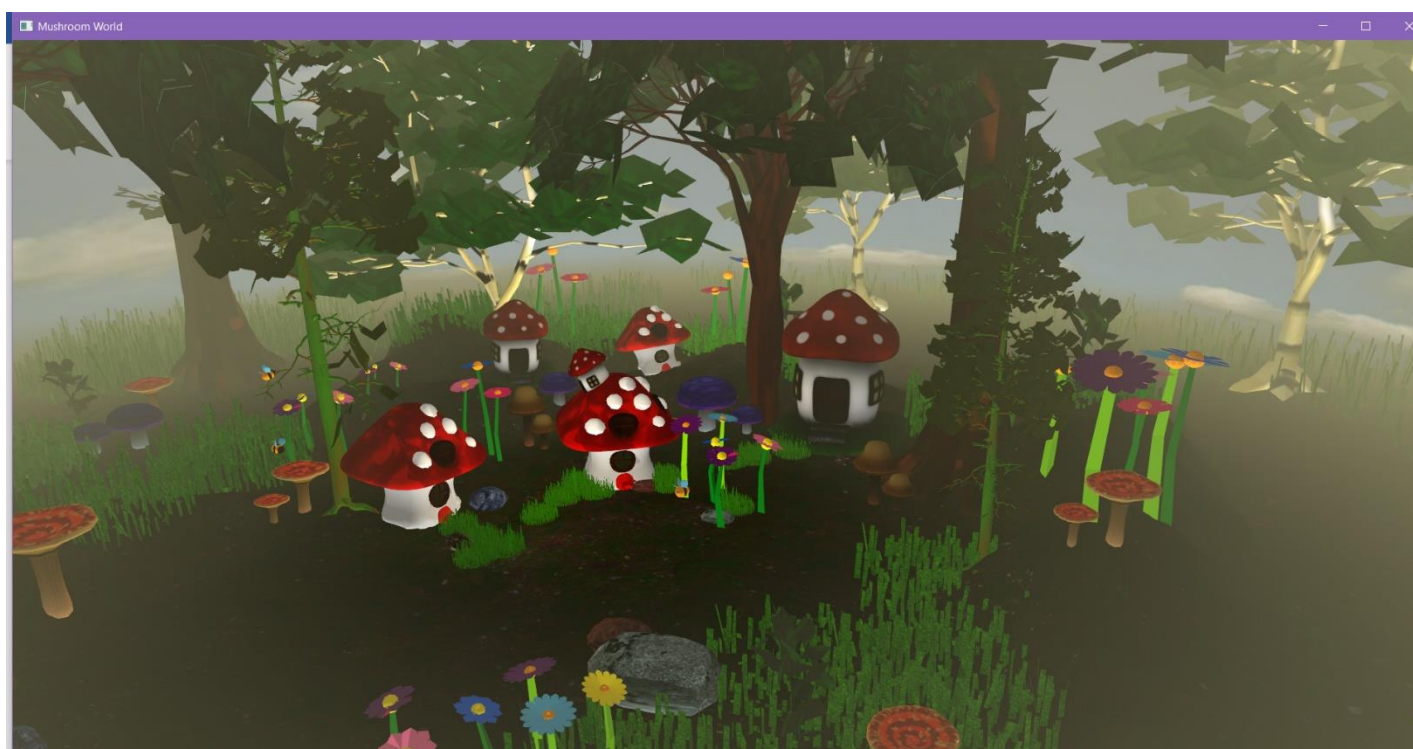


Figure 5 Scena văzută cu ceață





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

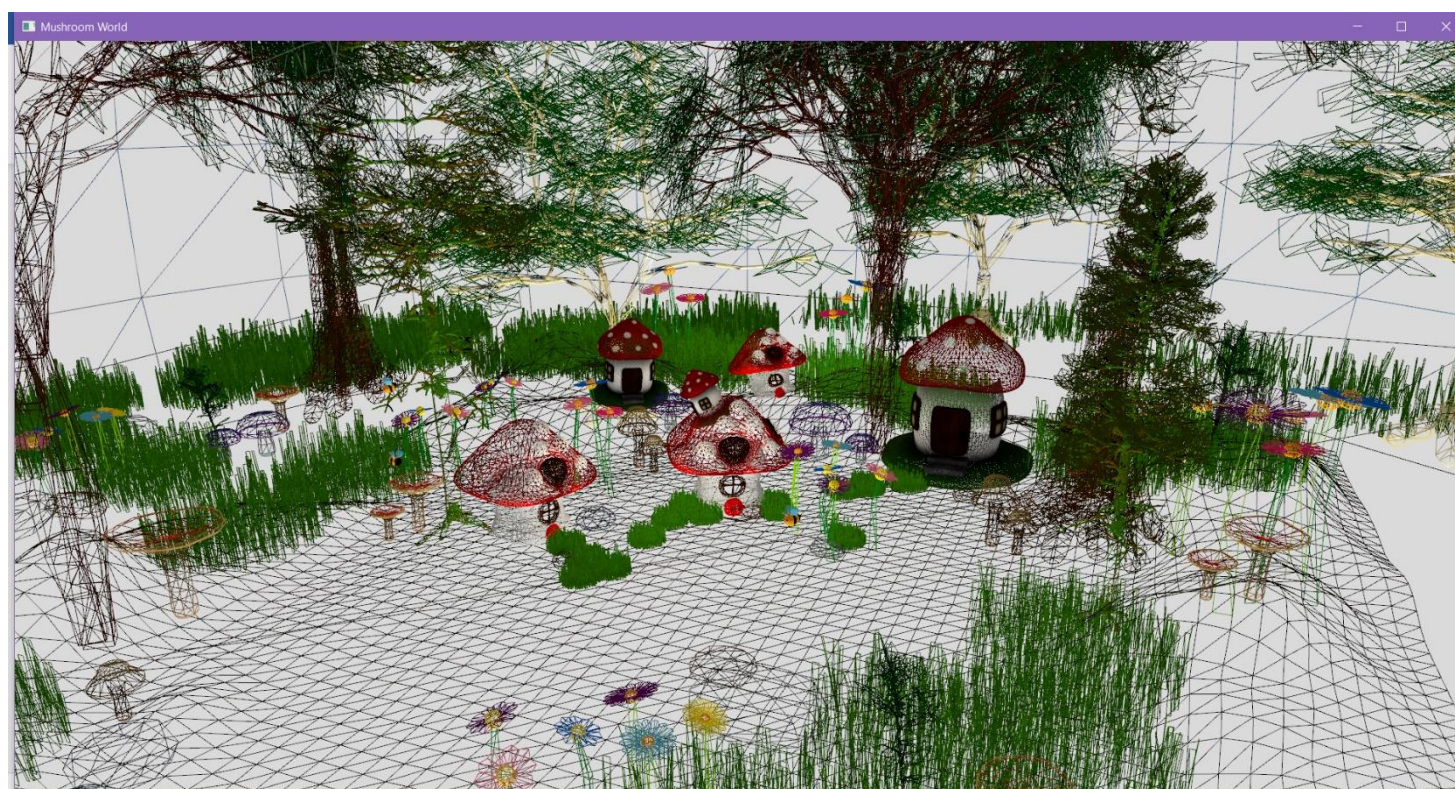


Figure 6 Modul wireframe



Figure 7 Modul polygonal



### 3. Detalii de implementare și algoritmi

Proiectul a fost implementat având ca bază codul sursă pus la dispoziție la laborator, au fost adăugate diferite funcții în main.cpp, create cu ajutorul informațiilor primite la laborator, și implementarea clasei Camera.cpp.

Principalele funcții utilizate pentru a ajunge la rezultatul final au fost pe rând:

- windowResizeCallback() folosită pentru a putea modifica dimensiunile scenei atunci când se modifică dimensiunea ferestrei.
- keyboardCallBack() funcție în care sunt implementate diferite comenzi date de la tastatură (escape, solid, wireframe, poligonal, smooth, ceață, zi/noapte, animații albine/castor)
- mouseCallBack() funcție implementată pentru control de camera de la mouse
- processMovement() funcție pentru mișcarea camerei de la tastatura (sus, jos, stanga dreapta, față, spate, rotire stanga/dreapta)
- initModels() și initShaders() funcții care inițializează cele trei modele din scenă (world, bee, beaver) și shaderele (basic.vert și basic.frag)
- initUniforms() funcție care creează matrice model pentru cele 3 obiecte, matrice view pentru cameră, matricea de proiecție, precum și variabilele pentru lumina direcțională și luminile de tip point, și ceața.
- renderWorld(), renderBee() și renderBeaver funcții care folosind shader-ul dat apelează draw pe fiecare din cele 3 obiecte ca mai apoi toate 3 funcțiile să fie apleate în funcția renderScene()
- main() funcție care unește toate celelalte funcții

#### 3.1 Algoritmi

Unul dintre algoritmii folosiți în cadrul acestui proiect este algoritmul de iluminare cu ajutorul căruia lumina se reflectă de pe suprafețe. Un alt algoritm implementat este cel pentru crearea ceții în scenă.

Deoarece scena a fost creată inițial în Blender și mai apoi exportată am ales să implementez un skyDome în loc de skyBox. De asemenea cu ajutorul Blender-ului am putut să observ coordonatele la care se află diferite obiecte și să modific coordonatele camerei.

##### 3.1.1 Soluții posibile

Pentru implementarea luminii în scenă se putea alege între diferite tipuri de lumina, direcțională, de tip point și spotlight, de asemenea, trebuiau implementate două surse



**UNIVERSITATEA TEHNICĂ**

DIN CLUJ-NAPOCA

diferite de lumină. De asemenea se putea alege între diferiți algoritmi precum generarea dinamică de obiecte, detecția coliziunilor, umbrele, ceață, ploaie, vânt.

### **3.1.2 Motivarea abordării alese**

Am ales să implementez două tipuri de lumini diferite, lumina principală de zi sau de noapte care acoperă toată scena este lumina direcțională ce poate fi schimbată de la tastatură și două lumini de tipul point light. Am considerat ca ar fi foarte sugestiv să folosesc aceste tipuri de lumini în căsuțele de ciuperci, care pe timp de noapte dau impresia că ar fi locuite, luminile de tip point având coordonate exact în interiorul celor doua ciupercuțe mai mari din față.

De asemenea, pentru un plus de realism am folosit algoritmul de creare a ceții deoarece am considerat că se potrivește foarte bine cu scenariul misterios a scenei din pădure.

## **3.2 Modelul grafic**

Modelul grafic regăsit în această scenă 3D cuprinde utilizarea unei iluminări ambientale, difuze și speculare, algoritmi implementați în fragment shader, pentru adăugarea diferitelor surse de lumină astfel încât să fie totul cât mai realist

## **3.3 Structuri de date**

Structurile de date folosite sunt cele specifice bibliotecii OpenGL: matrici, vectori de diferite dimensiuni, Camera, Shader, Model3D, Window, GLint, GLuint, GLfloat, GLboolean, bool, float, int.

## **3.4 Ierarhia de clase**

Proiectul conține mai multe clase, atât cpp, cât și hpp precum: clasa Camera, clasa Shader, Mesh, Model3D, stb\_image, tiny\_obj\_loader, Window și nu în cele din urmă clasa main în care se desfășoară principalul cod și în care s-a lucrat cel mai mult desigur în cadrul acestui proiect.

## **4. Manual de utilizare**

W – mișcarea în față a camerei

A – mișcarea în stânga a camerei





- S – mișcarea în spate a camerei
- D – mișcarea în dreapta a camerei
- Q – rotirea spre stânga a camerei
- E – rotirea spre dreapta a camerei
- Left Shift – mișcarea camerei în jos
- Space – mișcarea camerei în sus
- 1 – vizualizarea scenei în modul wireframe
- 2 – vizualizarea scenei în modul solid
- 3 – vizualizarea scenei în modul poligonal
- 4 – vizualizarea scenei în modul smooth
- 5 – scădere densitate ceață
- 6 – creștere densitate ceață
- 7 – scena pe timp de zi
- 8 – scena pe timp de noapte
- V – activare/dezactivare animație albine
- M – animație castor într-o direcție
- N – animație castor în altă direcție

## **5. Concluzii și dezvoltări ulterioare**

Scena 3D prezentată în proiectul de față este o scenă mică, dar bogată, cu multe obiecte și diferite texturi, însă există loc de dezvoltări ulterioare, precum adăugarea unor umbre pentru a sporii gradul de fotorealism, de asemenea, se poate mări dimensiunile scenei și pot adăuga animații mai realiste personajelor, precum și adăugarea coliziunilor.

## **6. Referințe**

[https://docs.google.com/document/d/1njtWPMmOQNlaD\\_z9ve8iPRUqQTwdIV\\_PO-NvPD0nOuM/edit](https://docs.google.com/document/d/1njtWPMmOQNlaD_z9ve8iPRUqQTwdIV_PO-NvPD0nOuM/edit) - pentru crearea mai ușoară a scenei în Blender

<https://www.youtube.com/watch?v=-NSBP5q8nNE> – pentru Blinn-Phong

<https://learnopengl.com/> - pentru documentația diferitelor funcții din biblioteca OpenGL

[www.turbosquid.com](http://www.turbosquid.com), [www.tf3dm.com](http://www.tf3dm.com), și <https://sketchfab.com/features/free-3d-models> pentru modelele 3D prezente în scenă.