Documentatie PG

Bretea Vlad Grupa 30237

Cuprins

1.Prezentarea temei

2.Scenariul

2.1.Descrierea scenei si a obiectelor

2.2.Functionalitati

3.Detalii implementare

3.1.Functii si algoritmi

3.1.1. Solutii posibile

3.1.2. Motivarea abordarii alese

3.2 Model grafic

3.3. Structuri de date

3.4. Ierarhia de clase

4.Prezentarea interfetei grafice utilizator/ manual de utilizare

5. Concluzii si dezvoltari ulterioare

6. Referinte

1.Prezentarea temei

Tema acestui proiect este crearea unei harti in OPENGL in care sa putem performa diferite actiuni: deplasare cu mouse, tastatura, crearea de umbre, ceata , etc. Acest proiect urmareste familiarizarea cu OpenGL, si biblioteci precum glfw,glm. Pe langa aceste biblioteci si OPENGL , as preciza ca in crearea proiectul am mai folosit Blender.  
2.Scenariul

2.1.Descrierea scenei si a obiectelor

Scena reprezinta o fortareata in desert , o colonie de oameni si-au stabilit aceasta locatie si au construit aceste cladiri pentru a trai in ele, dar si pentru a le oferi protectie. Obiectele prezente in aceasta scena sunt: diverse cladiri , plante, copaci.

2.2.Functionalitati

Functionalitatile acestui proiect sunt deplasarea prin scena folosind W A S D, rotirea camerei folosind E si Q.

3.Detalii de implementare

3.1.Functii si algoritmi

3.1.1.Solutii posibile

1.initShaders - este responsabila pentru incarcarea si initializarea shaderelor. Shader-ele dau utilizatorului control asupra fiecarui vertex si fragment. Spre exemplu lightShader.loadShader("shaders/lightCube.vert", "shaders/lightCube.frag");

lightShader.useShaderProgram();

Se ocupa de incarcarea unui shader pentru afisarea unui light cube, pentru ca dupa aceea sa fie activat folosind useShaderProgram().

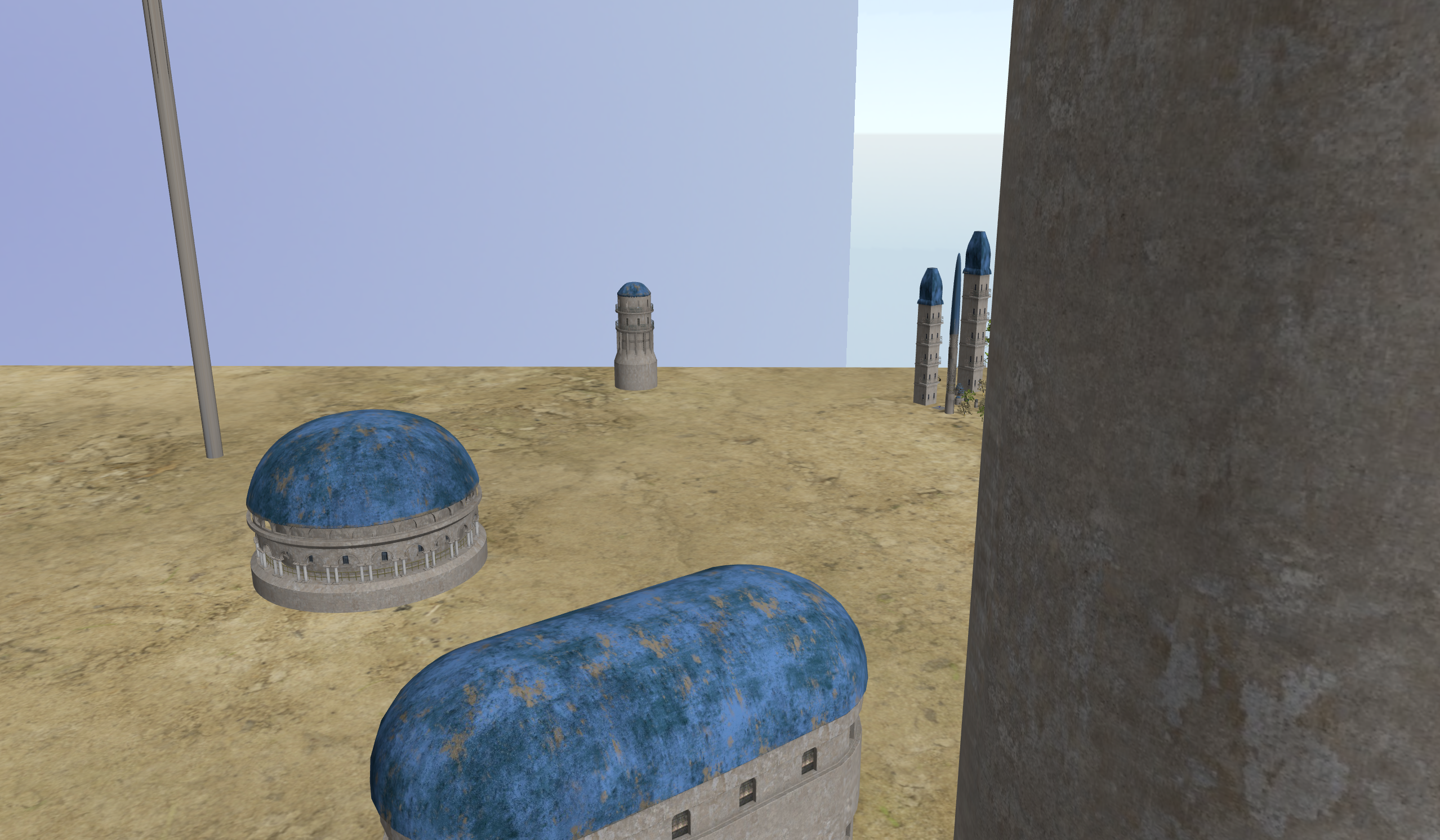
2.windowsResizeCallback - se asigura ca de fiecare data cand fereastra este modificata, diferite componente : framebuffer0ul, dimensiunile ferestrei, viewport si matricea de proiectie sunt updatate in mod corespunzator.

3.mouseCallback - urmareste miscarea mouse-ului si schimba orientarea camerei in mod corespunzator, permitand utilizatorului sa controleze view-ul camerei miscand mouse-ul

3.1.2.Motivarea alegerii

Am folosit aceste functii deoarece utilizatorul trebuie sa poata vedea intreaga harta , asta fiind posibil doar cu ajutorul mouseCallback-ului si windowsResize. De asemenea harta trebuie sa fie luminata, ceea ce a fost posibil doar cu ajutorul initShader-ului.

3.2.Modelul grafic



Pentru a crea harta, am folosit shadere si obiecte pe care le-am gasit pe diferite site-uri pe care le-am texturat si pozitionat pe un plan in blender si apoi le-am exportat ca si obj.

3.3.Structuri de date

Structurile de date folosite sunt: vectori, faces folosit pentru randarea skybox-ului, array-uri, pressedKeys folosit pentru a urmari input-ul de la tastatura, structuri de date scalare si booleene: retina\_height,retina\_width,previousX,cameraMove, etc. Se mai folosesc Enum-uri si constante: GL\_COLOR\_BUFFER\_BIT, GL\_DEPTH\_BUFFER\_BIT, clase customizate: gps::Camera, gps::Window, gps::SkyBox.

3.4.Ierarhia de clase

In principal , relatia exista intre clasele din proiect este una de compozitie, obiectele unei clase sunt folosite in alta clasa pentru a contribui la functionalitatea programului.

gps:: Window – contine functionalitatile pentru management-ul ferestrei.

gps:SkyBox: se ocupa de incarcarea si randarea skybox-ului.

gps::Camera – reprezinta camera scenei

gps::Shader – se ocupa de programul de shadere, compileaza si incarca vertex si fragment shader-ul.

gps::Model3D –se ocupa de incarcarea si randarea modelelor 3D.

4.Prezentarea interfetei grafice/manual de utilizare

Pentru a te deplasa prin interfata grafica se folosesc tastele : W A S si D. Pentru a te roti se foloseste Q sau E, iar pentru a misca camera in sus sau in jos se poate folosi mouse-ul.

5.Concluzii si dezvoltari ulterioare

In concluzie , acest proiect m-a ajutat sa inteleg cum se creaza o harta virtuala , cat de mult efort trebuie pus in partea de creare a obiectelor si texturare, precum si pozitionare a acestor obiecte in pozitii potrivite. Dezvoltarile ulterioare cuprind: adaugarea de umbre si ceata .

6.Referinte

[Creating animations using Transformations in OpenGL - GeeksforGeeks](https://www.geeksforgeeks.org/creating-animations-using-transformations-opengl/)

[OpenGL program for simple Animation (Revolution) in C - GeeksforGeeks](https://www.geeksforgeeks.org/opengl-program-simple-animation-revolution-c/)