

Documentatie proiect SPG

Joc – Mediu maritim

Student: Florea Gabriel-Alin

Grupa: 30236

An: 3

Sectia: Calculatoare Romana

Profesor: Adrian Sabou

Cuprins

- 1.) Prezentarea temei
- 2.) Scenariul
 - a. Scena si obiectele
 - b. Functionalitati
- 3.) Detalii de implementare
 - a. Functii si algoritmi
 - i. Solutii posibile
 - ii. Motivarea abordarii alese
 - b. Modelul grafic
 - c. Structuri de date
 - d. Ierarhia de clase
- 4.) Prezentarea interfetei utilizator si manual de utilizare
- 5.) Concluzii si dezvoltari ulterioare
- 6.) Referinte

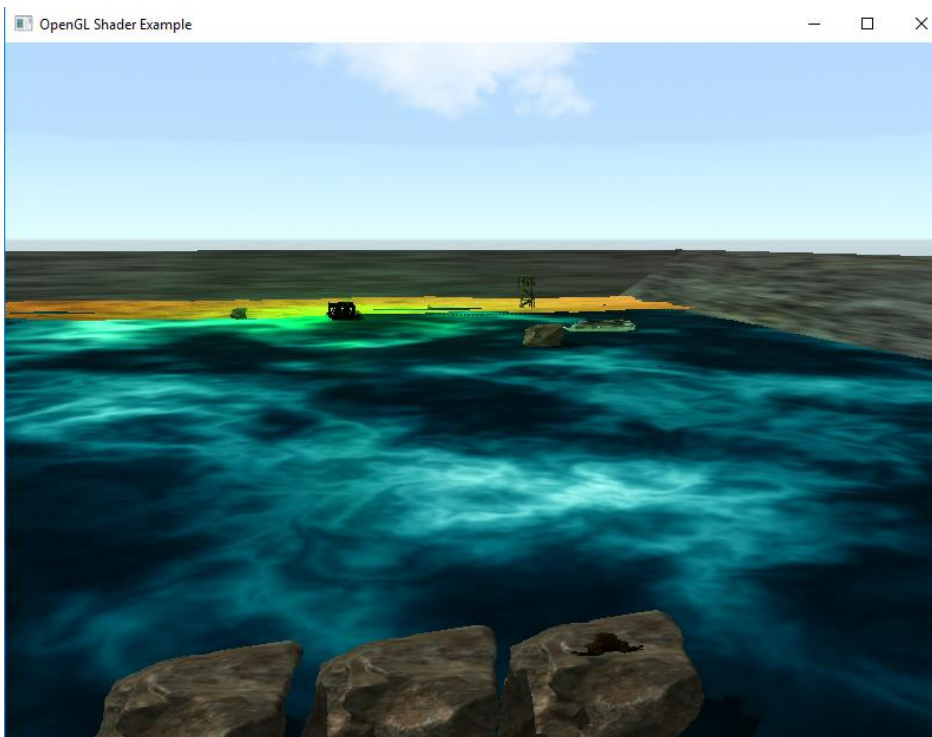
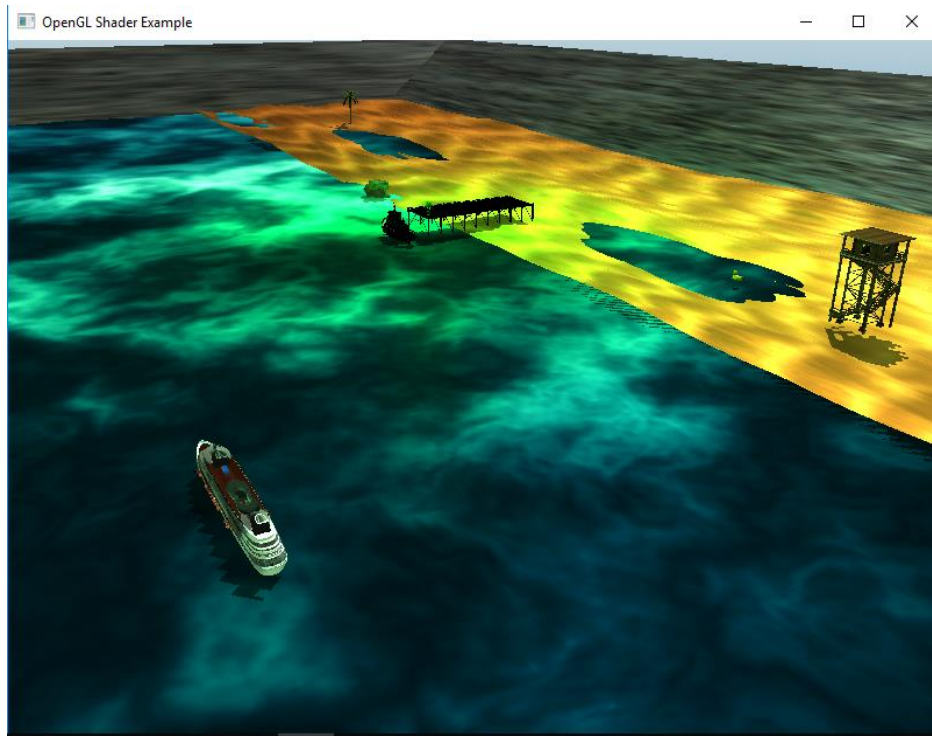
1.) Prezentarea temei

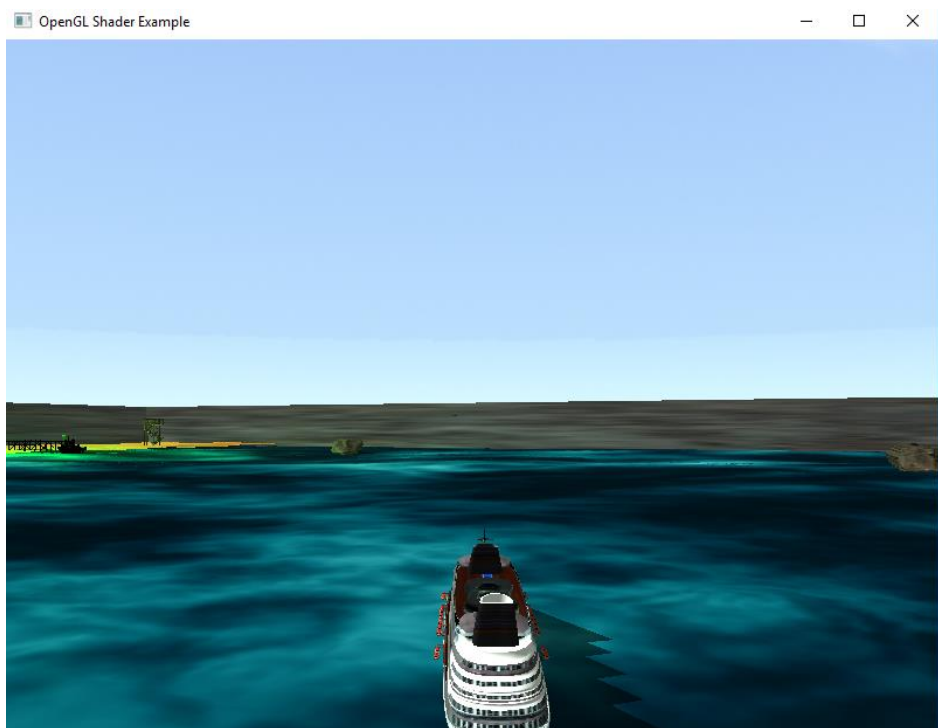
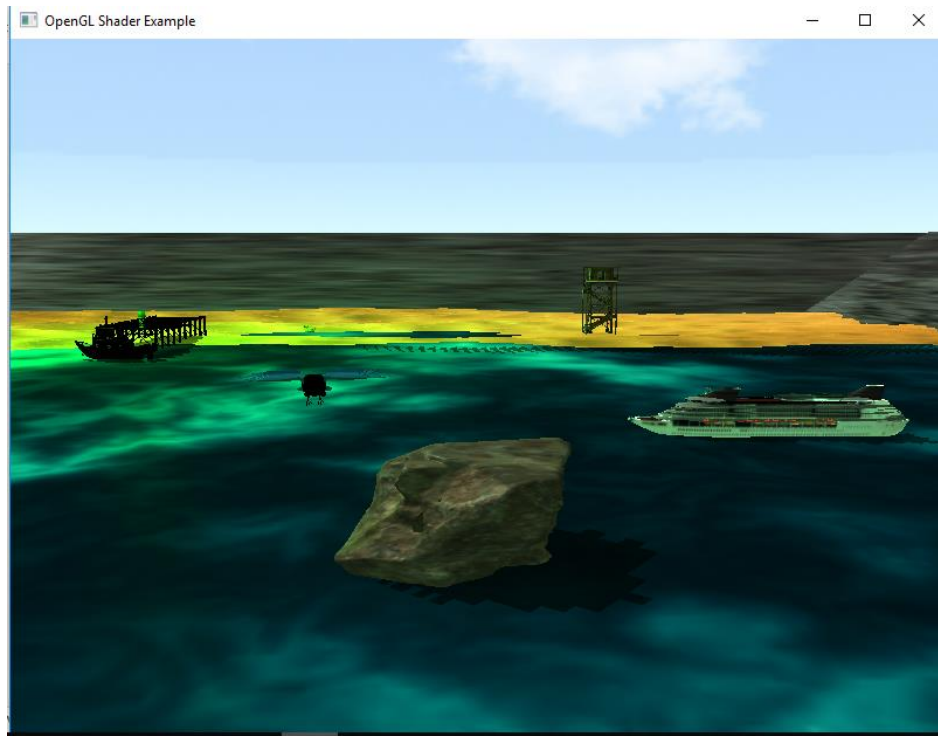
Tema acestui proiect este crearea unei aplicatii grafice interactive, fotorealiste, utilizand OpenGL si celelalte librarii prezentate in cadrul sesiunilor de laborator, aplicatie in care este posibila vizualizarea scenei cu ajutorul comenzilor mouse si tastatura, cat si cu ajutorul unei animatii de prezentare predefinite. Scena, care reprezinta cadrul de desfasurare a jocului permite operatii de scalare, translatie si rotatie. In cadrul acestui joc s-au folosit doua tipuri surse de lumina, cea directionala si cea punctiforma. Scena de obiecte poate fi vizualizata in diferite moduri, si anume wireframe, prin puncte si modul de vizualizare normal a obiectelor. Toate obiectele din care este compusa scena au aplicate texture, care creeaza fotorealism aplicatiei. Miscarea unui anumit obiect, vizualizarea umbrelor diferitelor obiecte, cat si interactiuni cu obiectele (collision detection). De asemenea jocul permite simularea fenomenului de ceata si mutarea sursei de lumina, astfel incat se poate simula si efectul de zi/noapte.

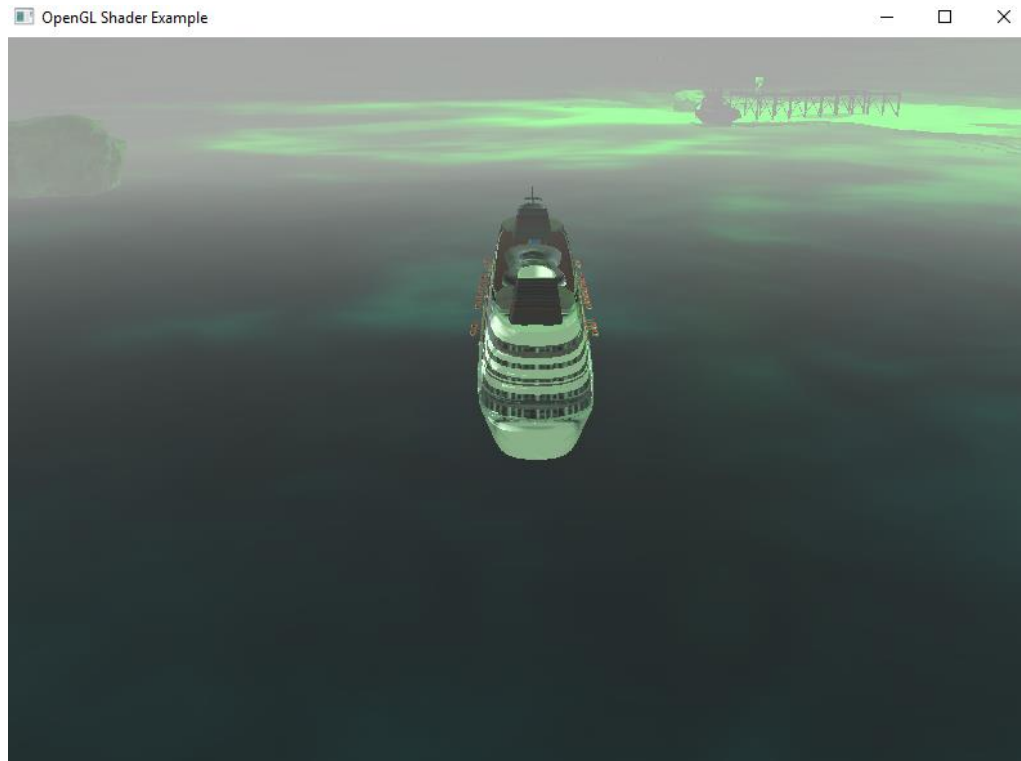
2.) Scenariul

a. Scena si obiectele

Scenariul ales pentru acest proiect este un mediu maritim, reprezentat de o suprafata de apa, o plaja alaturata, nava care se poate misca, un turn de supraveghere a activitatii maritime, un ponton, o barca langa acel ponton, un emitator de lumina de culoare verde aflat pe ponton, vizibil chiar si pe timp de ceata sau noapte, niste pereti de delimitare vizuala a scenei de obiecte, o rata care se plimba pe o bucata de apa, o broasca care sta la soare pe o stanca, un copac, niste stanci si o pasare care zboara deasupra apei. Baltile de pe plaja au fost obtinute prin utilizarea unei mici diferente de translatie pe axa y, intre mare si plaja. De asemenea, s-a folosit un skybox care se potriveste cu scena de obiecte, pentru a imbunatatii fotorealismul scenei.







b. Functionalitati

Printre functionalitatile acestei aplicatii se numara posibilitatea de a vizualiza scena cu ajutorul mouse-ului si tastaturii, de a misca sursa de lumina pe doua axe, cu ajutorul tastaturii, pentru a putea vizualiza mai bine umbrele unui anumit obiect, cat si pentru a putea simula efectul de zi/noapte, si de a misca nava principala cu ajutorul sagetilor. Printre functionalitati se numara de asemenea, miscarea ratei pe balta, zborul pasarii deasupra apei, respectiv detectia coliziunilor navei cu obiectele intalnite. Aplicatia permite de asemenea centrarea camerei pe barca, pe pozitia standard de vizualizare pentru conducerea barcii, la apasarea unei taste. Animatia de prezentare poate fi pornita de asemenea cu ajutorul unei taste, putand fi intrerupta in cazul in care utilizatorul nu doreste sa astepte finalizarea ei. Aplicatia permite, pe langa vizualizarea normala a scenei de obiecte si vizualizarea acestora in modul wireframe, tot cu ajutorul unor taste dedicate. De asemenea, efectul de ceata poate fi simulat, tot prin utilizarea tastaturii in acest scop.

3.) Detalii de implementare

Aceasta aplicatie grafica interactiva a fost realizata in mediul Visual Studio, utilizand biblioteci dedicate graficii pe calculator, cum ar fi: OpenGL, GLFW, GLEW, GLM.

a. Functii si algoritmi

Dintre posibilele abordari alese pentru implementarea diverselor functii necesare acestei aplicatii, am ales acele abordari, care au fost prezentate in cadrul sesiunilor de laborator si a cursurilor la materia Sisteme de Prelucrare Grafica, intrucat, am considerat ca fiind cele mai de baza si importante abordari care trebuie cunoscute.

Pentru a putea utiliza tastatura si mouse-ul, am inclus in codul aplicatiei niste functii de callback pentru aceste periferice.

Detectia coliziunilor am realizat-o, utilizand o implementare minimala a tehnicii axis-aligned bounding box, folosita in grafica pentru detectia aproximativa a coliziunilor dintre obiecte.

Funcția care implementează animația de prezentare include o logică de bază, care permite astfel, realizarea acestei animații de prezentare, mutând poziția camerei și efectuând rotații necesare.

Funcția de procesare a mișcării realizează maparea comenzilor de la tastatură la valorile unor variabile. De asemenea, tot în această funcție se calculează mișcarea obiectelor, a navei, a ratei și a pasării. Mișcarea acestora e calculată folosind funcțiile trigonometrice sinus și cosinus.

Funcția followShip, permite camerei să urmărească nava, pentru a putea vedea exact modul de mișcare a acesteia.

În clasa Camera, funcțiile pentru mișcarea și rotirea camerei, alături de limitările impuse mișcării camerei, relative la un anumit spațiu care poate fi vizualizat, permit o vizualizare bună a scenei de obiecte.

Iluminarea scenei, a fost realizată (în shader) cu ajutorul modelului de iluminare Blinn-Phong. Au fost utilizate două tipuri de lumină, cea direcțională cu componentele (ambientală, difuză și speculară), alături de cea punctiformă cu aceleași componente, însă prezentând o atenuare odată cu distanța. Umbrele obiectelor au fost adăugate

folosind tehnica Shadow Mapping, cu metoda depth-bufferului. De asemenea a fost adaugat si efectul de ceata in scena.

b. Structuri programului

Programul acesta, fiind un program OpenGL, respecta o structura standard a programelor de acest tip. Astfel programul creeaza o fereastra cu anumite caracteristici, in interiorul careia deseneaza obiectele din scena. Pentru a putea desena scena in mod corespunzator, aplicatia se foloseste de shadere.

c. Modelul grafic

Obiectele utilizate in cadrul scenei au fost preluate de pe site-uri care ofera obiecte 3D in mod gratuit. S-au folosit doar obiecte .obj, impreuna cu fisierul .mtl atasat si cu texturile obiectului respectiv.

Pozitia, marimea si orientarea obiectelor a fost transmisa catre shadere cu ajutorul unor functii OpenGL dedicate pentru aceasta, iar punerea in scena a obiectelor s-a realizat cu functia draw, pusa la dispozitie de OpenGL.

d. Structuri de date

Acest program utilizeaza structuri de date clasice (variabile simple de tip: int, std::vector, float, bool), dar utilizeaza de asemenea si variabile specifice OpenGL, cum ar fi: GLFWwindow, glm::mat3, glm::mat4, glm::vec3, glm::vec4, GLfloat, gps::Model3D, gps::Shader, GLuint, gps::Skybox, GLenum.

In shadere se utilizeaza si variabile de tip sampler2D.

Comunicarea intre programul de baza si shadere se realizeaza si prin intermediul variabilelor uniform.

e. Ierarhia de clase

Acest program este structurat pe mai multe clase, insotite de fisiere header. Dintre clase, mai semnificative sunt urmatoarele:

- Camera – care se ocupa de functionalitatile camerei
- Model3D – care se ocupa de interpretarea obiectelor 3D
- Shader – care permite utilizarea shaderelor
- SkyBox – care permite adaugarea unui skybox aplicatiei
- OpenGL_4_Application_VS2015 – care e clasa de baza a aplicatiei si care utilizeaza toate clasele mai sus amintite, impreuna cu biblioteci specializate pentru grafica.

4.) Prezentarea interfetei utilizator si manual de utilizare

Modul in care utilizatorul aplicatiei poate interactiona cu aceasta este unul cat se poate de intuitiv. Odata cu rulara aplicatiei, utilizatorul este postat in pozitia de vizualizare a navei, din pozitia de conducere a acesteia. Utilizand sagetile, utilizatorul poate deplasa dupa dorinta nava, iar in cazul in care nava va intra in coliziune cu vreun obiect, se va detecta aceasta coliziune si utilizatorul nu va putea continua deplasarea in acea directie. Utilizatorul poate misca camera de vizualizare dupa dorinta, in limitele impuse din cod, cu ajutorul mouse-ului, a scroll-ului de la mouse si cu ajutorul tastelor W,A,S,D. Pentru a rula animatia de prezentare a scenei, utilizatorul va apasa tasta R, iar pentru a opri derularea prezentarii va apasa tasta Z. Daca protagonistul doreste sa se posteze din nou in pozitia barcii, e suficient sa apese tasta T. In cazul in care utilizatorul doreste vizualizarea scenei de obiecte in modul wireframe, atunci el va apasa tasta F, pentru a vizualiza scena sub forma de puncta va apasa H, iar pentru a reveni la modul normal de vizualizare va apasa tasta G. Pentru a modifica pozitia sursei de lumina pe scena, se vor utiliza tastele J si L, iar pentru a modifica pozitia sursei de lumina pentru a simula efectul de zi/noapte, se vor utiliza tastele I si P. Adaugarea efectului de ceata se va face prin mentinerea apasata a tastei B, pana la obtinerea intensitatii dorite a fenomenului, iar diminuarea acestui efect, se poate realiza cu ajutorul tastei N.

Pe langa toate acestea, utilizatorul aplicatiei se mai poate bucura si de observarea pasarii care zboara deasupra apei, cat si a ratei care se plimba pe apa, admirand astfel un peisaj maritim, deosebit de frumos, o adevarata oaza de liniste.

5.) Concluzii si dezvoltari ulterioare

Acest proiect, reprezentat de acest joc, este doar o schita a ceea ce un astfel de joc ar putea fi. Proiectul de fata, reprezinta mai degraba rezultatul aplicarii intr-un timp limitat a unor notiuni invatate la materiile de grafica pe calculator, in cadrul Universitatii Tehnice Cluj-Napoca. Astfel, acest proiect permite o multime de posibile dezvoltari ulterioare, incepand de la calitatea scenei, introducerea unui scop bine definit in cadrul jocului, nu doar free running, imbunatatiri de eficienta, de efecte, utilizarea unor tehnici si algoritmi superiori calitativ, si lista ar putea continua. A face aceasta aplicatie compatibila cu mai multe device-uri, inclusive cu dispozitive mobile, ar fi cu siguranta o imbunatatire semnificativa a aplicatiei.

Acest proiect, a constituit o oportunitate deosebita de invatare si aplicare a notiunilor practice, de baza, invatate la materiile de grafica pe calculator, fapt care a facut din acest proiect o adevarata provocare pentru mine personal si o oportunitate de a ma documenta si de a studia o multime de notiuni si de concepte necesare pentru realizarea acestui proiect.

6.) Referinte

- <https://moodle.cs.utcluj.ro/> - cursurile si laboratoarele de SPG
- <https://free3d.com/>
- <https://www.turbosquid.com/Search/3D-Models/free>
- <https://learnopengl.com/Lighting/Basic-Lighting>
- <https://learnopengl.com/Lighting/Light-casters>
- <https://learnopengl.com/In-Practice/2D-Game/Collisions/Collision-detection>