|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Proiect OpenGL: Planeta | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Numele studentului Calbeaza Mihaela-Maria  Grupa:6  **Universitatea Tehnica Cluj-Napoca**  **Facultatea de Automatica si Calculatoare**  **Prof. coordonator:** Assist. Eng. Dan Gota |  | | | |
|  | | | |  |  | | | |
|  | | |  | | |  | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  |  | | | |  |
|  | CUPRINS | | | | | | |  |
|  |  | |  |  | | | |  |
|  |  | | | |  |  | |  |
|  |  | Descriere proiect ………………………………...3  Screenshot proiect…………………………………………3 | | |  |  |  |  |
|  | | | | |
|  |  |  |  |
|  |  |



# Descriere proiect

**Acest proiect reprezinta o planeta cu 3 inele care se misca pe orbita acesteia.**

**Acesta este un proiect OpenGL, scris in limbajul C++.**

**Inelele planetei au fost realizate cu ajutorul functiei “glutSolidTorus”, iar pentru aplatizarea lor am ales 2 laturi. Planeta a fost realizata cu ajutorul clasicei functii “glutSolidSphere”. Miscarea inelelor are la baza functia “glRotatef” iar unghiurile de rotatie sunt stocate in 3 variabile.**

**Cerinte hardware:**

* **PC cu procesor Intel generaţia a treia Core i3.0 (sau AMD, echivalent)**
* **8 GB RAM. În funcţie de dimensiunile şi complexitatea proiectului, se recomandă suplimentarea RAM.**
* **Minimum 50 GB spaţiu liber pe hard disk.**
* **Placă video compatibilă Windows (1 GB memorie video recomandată) .**

**A picture containing basketball, light, drawing

Description automatically generated**

**//Calbeaza Mihaela Maria**

**#include <windows.h>**

**#include <GL/glut.h>**

**void activitate();**

**GLfloat xRotatie, yRotatie, zRotatie; //unghiuri de rotatie**

**void display()**

**{**

**glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); //curata ecranul si buffer-ele**

**glMatrixMode(GL\_MODELVIEW); //specifica care matrice este utilizata**

**glLoadIdentity(); //inlocuieste matricea curenta cu o matrice de identitate**

**glPushMatrix(); //impinge si deschide stiva actuala a matricii**

**glRotatef(xRotatie, 1.0, 0.0, 0.0); //inmulteste matricea actuala cu o matrice de rotatie xRotatie, yRotatie, zRotatie fiind unghiurile de rotatie**

**glRotatef(yRotatie, 0.0, 1.0, 0.0);**

**glRotatef(zRotatie, 0.0, 0.0, 1.0);**

**glScalef(1.0, 1.0, 1.0); //inmulteste matricea curenta cu o matrice de scalare**

**glutSolidTorus(0.02, 1.5, 3, 100); //creeaza inelele planetei**

**glRotatef(xRotatie, 0.5, 0.0, 0.0);**

**glRotatef(yRotatie, 0.0, 0.5, 0.0);**

**glRotatef(zRotatie, 0.0, 0.0, 0.5);**

**glutSolidTorus(0.2, 1.2, 2, 75);**

**glRotatef(xRotatie, 1.5, 0.0, 0.0);**

**glRotatef(yRotatie, 0.0, 1.5, 0.0);**

**glRotatef(zRotatie, 0.0, 0.0, 1.5);**

**glutSolidTorus(0.12, 0.85, 2, 50);**

**glutSolidSphere(0.5, 30, 50);**

**glPopMatrix(); //sfera a fost plasa in "interiorul" inelelor, aceeasi origine**

**glutSwapBuffers(); //se foloseste metoda de intershimbare a bufferelor**

**glFlush();**

**}**

**void activitate(void)**

**{**

**xRotatie += 0.02;**

**yRotatie += 0.02;**

**zRotatie += 0.02;**

**display();**

**}**

**void remodelare(GLint w, GLint h)**

**{**

**glViewport(0, 0, w, h); //seteaza punctul de vedere**

**glMatrixMode(GL\_PROJECTION); //matricea curenta devine matricea de proiectie**

**GLfloat aspect = GLfloat(w) / GLfloat(h);**

**glLoadIdentity();**

**if (w <= h)**

**{**

**// daca lățimea este mai mică, inaltimea se intinde**

**glOrtho(-2.5, 2.5, -2.5/aspect, 2.5/aspect, -10.0, 10.0);**

**}**

**else**

**{**

**// daca inaltimea este mai mică, latimea se intinde**

**glOrtho(-2.5\*aspect, 2.5\*aspect, -2.5, 2.5, -10.0, 10.0);**

**}**

**}**

**void styles() //Styling-ul Planetei cu scopul de a creea un efect realist**

**{**

**GLfloat negru[] = { 0.0, 0.0, 0.0, 1.0 };**

**GLfloat rozaliu[] = { 1.0, 1.0, 0.0, 1.0 };**

**GLfloat roz[] = { 1.0, 0.0, 1.0, 0.0 };**

**GLfloat alb[] = { 0.0, 1.0, 1.0, 1.0 };**

**GLfloat directie[] = { 1.0, 1.0, 1.0, 0.0 };**

**glMaterialfv(GL\_FRONT, GL\_AMBIENT\_AND\_DIFFUSE, roz); //iluminarea obiectului pe partea din fata folosind lumina ambientala**

**glMaterialfv(GL\_FRONT, GL\_SPECULAR, alb);**

**glMaterialf(GL\_FRONT, GL\_SHININESS, 30);**

**glLightfv(GL\_LIGHT0, GL\_AMBIENT, negru); //seteaza lumina ambientala-negru(palida)**

**glLightfv(GL\_LIGHT0, GL\_DIFFUSE, rozaliu); //seteaza lumina difuza: rozaliu-spre mov**

**glLightfv(GL\_LIGHT0, GL\_SPECULAR, alb); //da stralucire obiectului**

**glLightfv(GL\_LIGHT0, GL\_POSITION, directie);**

**glEnable(GL\_LIGHTING); // openGL "afla" de conceptul de lumina**

**glEnable(GL\_LIGHT0); // se aprinde sursa de lumină cu numărul 1**

**glEnable(GL\_DEPTH\_TEST); // rasterizare si adancimea**

**}**

**// Configurează atribute globale: culoarea clară și culoarea desenului**

**int main(int argc, char\*\* argv)**

**{**

**glutInit(&argc, argv); //se initializeaza libraria glut-ului**

**glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);**

**glutInitWindowSize(800, 600);**

**glutInitWindowPosition(80, 80);**

**glutCreateWindow("Planeta");**

**styles(); // functie pentru stilizare este chemata**

**xRotatie=40;**

**yRotatie=50;**

**zRotatie=60;**

**glutDisplayFunc(display); //seteaza afisajul din functia display**

**glutReshapeFunc(remodelare);**

**glutIdleFunc(activitate);**

**glutMainLoop();**

**return 0;**

**}**