****

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII**

**AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică şi Microelectronică**

**Departamentul Informatică şi Ingineria Sistemelor**

Chistol Maxim

Gr.IA-233

**Raport**

**pentru lucrarea de laborator Nr.6**

***la cursul de “Grafica pe Calculator”***

Verificat:

**Mariana Osovschi,** *Lector Universitar,*

Facultatea FCIM,

**Chișinău – 2024**

**Tema:** Transformări 3D in dinamica

**Scopul lucrării:**

Obținerea cunoștințelor practice în sinteza scenelor grafice 3D dinamice, utilizând funcțiile standard de translație, rotație și scalare din biblioteca p5.js.

**Sarcina lucrării:**

1. Elaborați un program pentru sinteza unei scene 3D dinamice utilizând funcțiile standard de translație, rotație și scalare din biblioteca p5.js.

2. Elaborați un program care creează o scenă 3D dinamică conform variantei indicate în tabel Pentru crearea scenei pot fi utilizate obiecte grafice 3D existente în repozitoriul 3D.

3.Dinamice.

**Codul:**

**let fr = 30;**

**let angle = 0;**

**// Variabile pentru animație**

**let spherePos = 250;**

**let boxPos = -30;**

**let ellipsoidPos = -100;**

**let conePos = -130;**

**let torusPos = -130;**

**let boxScale = 1; // Variabila pentru controlul dimensiunii cubului**

**let scaleFactor = 0.02; // Viteză de schimbare a dimensiunii cubului**

**function preload() {**

**base = loadModel('f209d771-a3a0-459d-a19b-6d5f814c8ee5.obj');**

**}**

**function setup() {**

**createCanvas(800, 800, WEBGL);**

**normalMaterial();**

**frameRate(fr);**

**angleMode(DEGREES);**

**}**

**function draw() {**

**angle = 45 \* sin(millis() / 50);**

**orbitControl();**

**background(50);**

**noStroke();**

**push();**

**scale(0.1);**

**translate(0, 0, -4000);**

**model(base);**

**pop();**

**// Animăm sfera cu o funcție sinusoidală și rotație constantă**

**animateSphere();**

**// Animăm cubul cu o funcție de mărire și micșorare**

**animateBox();**

**// Animăm elipsoidul cu o funcție de tip săritură**

**animateEllipsoid();**

**// Animăm conul cu o funcție de tip oscillare**

**animateCone();**

**// Animăm torusul cu o funcție liniară inversă**

**animateTorus();**

**}**

**// Funcție pentru animarea cubului cu o funcție de scalare**

**function animateBox() {**

**// Schimbă dimensiunea cubului în funcție de timp folosind o funcție sinusoidală**

**boxScale = 1 + 0.5 \* sin(frameCount \* scaleFactor); // Cubul se va scala între 50% și 150% din dimensiunea sa originală**

**push();**

**translate(boxPos, 10, -630); // Poziția cubului**

**scale(boxScale); // Scalarea cubului**

**box(50); // Desenarea cubului**

**pop();**

**}**

**// Funcție pentru animarea elipsoidului cu o funcție de tip săritură**

**function animateEllipsoid() {**

**let ellipsoidSpeed = 0.8;**

**ellipsoidPos += ellipsoidSpeed;**

**let zPos = -300 + 100 \* floor(frameCount / 50 % 2);**

**push();**

**translate(ellipsoidPos, 40, zPos);**

**ellipsoid(30, 50, 40);**

**pop();**

**}**

**// Funcție pentru animarea conului cu o funcție de tip oscillare**

**function animateCone() {**

**push();**

**translate(conePos, 227, 170);**

**cone(60, 35);**

**pop();**

**}**

**// Funcție pentru animarea torusului cu o funcție liniară inversă**

**// Funcție pentru animarea torusului cu o funcție liniară inversă**

**function animateTorus() {**

**let torusSpeed = 0.5;**

**torusPos -= torusSpeed;**

**let scaleFactor = 1 + 0.5 \* sin(frameCount \* 0.02); // Factorul de scalare pentru animație**

**push();**

**translate(torusPos, 300, 170);**

**scale(scaleFactor); // Aplicăm scalarea torusului**

**torus(30, 40);**

**pop();**

**}**

**// Funcție pentru animarea sferei cu o funcție sinusoidală și rotație constantă**

**function animateSphere() {**

**let sphereSpeed = 0.5;**

**spherePos += sphereSpeed;**

**let yPos = 500 + 100 \* sin(frameCount \* 0.1);**

**let rotationSpeed = 0.5; // Viteză de rotație**

**push();**

**rotateY(frameCount \* rotationSpeed); // Rotire în jurul axei Y**

**translate(spherePos, yPos, -70);**

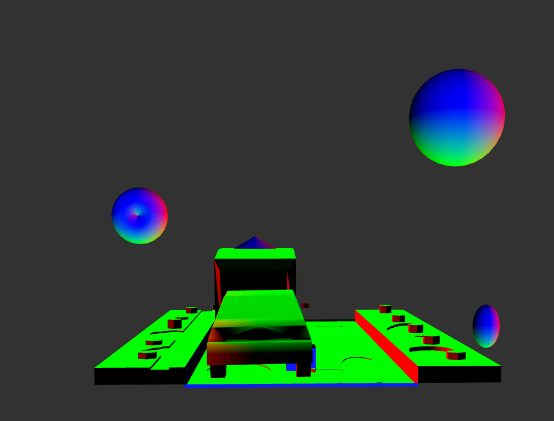
**scale(3.5);**

**sphere(40);**

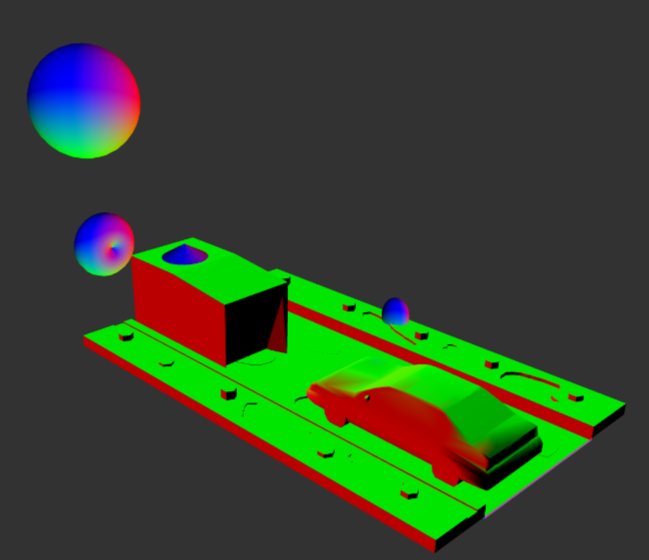
**pop();**

**}**

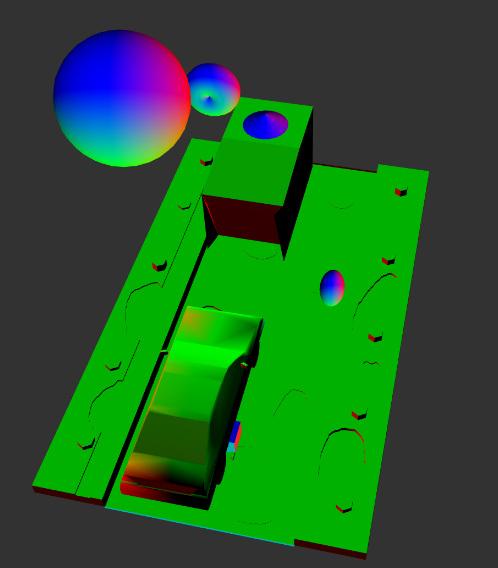
**Captura de ecran:**

****

**Fig.1**

****

**Fig.2**



**Fig.3**

**Concluzie:**

În cadrul acestei lucrări, am obținut cunoștințe practice în sinteza scenelor grafice 3D dinamice utilizând funcțiile standard de translație, rotație și scalare din biblioteca p5.js. Prin implementarea a două programe distincte, am demonstrat abilitatea de a manipula obiecte grafice într-un spațiu tridimensional pentru a crea scene dinamice și interactive.

Prin prima sarcină, am elaborat un program simplu care utilizează funcțiile de translație, rotație și scalare pentru a crea o scenă 3D dinamică. Acest exercițiu ne-a ajutat să înțelegem modul în care aceste funcții pot fi utilizate pentru a manipula poziția, orientarea și dimensiunile obiectelor într-un spațiu 3D.

În cea de-a doua sarcină, am creat o scenă 3D dinamică conform cerințelor specificate în tabelul 4.1. Utilizând obiecte grafice 3D existente în repozitoriul 3D și folosind funcțiile de translație, rotație și scalare, am reușit să generăm o scenă complexă și captivantă.

În timpul elaborării acestor programe, am întâlnit diverse provocări și am fost nevoiți să aplicăm cunoștințele teoretice în practică pentru a obține rezultatele dorite. Am dezvoltat o mai bună înțelegere a conceptelor de bază legate de grafica 3D și am câștigat experiență în utilizarea bibliotecii p5.js pentru a crea scene dinamice și interactive.

În concluzie, această lucrare ne-a oferit oportunitatea de a aplica cunoștințele teoretice într-un context practic și de a dezvolta abilități practice în sinteza scenelor grafice 3D dinamice. Am învățat să manipulăm obiecte grafice într-un spațiu tridimensional și să creăm scene captivante și interactive folosind biblioteca p5.js.