****

**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 modelarea proceselor 3D dinamice, utilizând funcțiile standard de translație, și rotație din biblioteca p5.js.**

**Sarcina lucrării:**

**1. Elaborați un program pentru modelarea unui proces fizic utilizând funcțiile standard de translație, și rotație din biblioteca p5.js.**

**2. Elaborați un program care creează o scenă 3D de modelare a proceselor fizice conform variantei indicate în tabelul 6.**

**1.Pentru crearea scenei pot fi utilizate obiecte grafice 3D existente în repozitoriul 3D.**

**Codul:**

**let t = 0; // Parametrul timpului**

**let dt = 0.2; // Pasul de timp**

**let trackPoints = []; // Punctele traseului**

**let coasterWidth = 7; // Lățimea roller coaster-ului**

**const rollerCoasterRadius = 10; // Raza roller coaster-ului**

**let coasterHistory = []; // Istoricul pozițiilor roller coaster-ului**

**function setup() {**

**createCanvas(800, 600, WEBGL);**

**// Adăugăm punctele pentru traseu**

**for (let x = -600; x <= 700; x += 10) {**

**let y = map(cos(x \* 0.05), -1, 1, -100, 100);**

**trackPoints.push(createVector(x, y, 0));**

**}**

**}**

**function draw() {**

**background(220);**

**orbitControl(); // Controlul camerelor**

**// Desenăm traseul**

**drawTrack(trackPoints);**

**// Simulăm mișcarea roller coaster-ului pe traseu**

**simulateCoaster(trackPoints);**

**// Desenăm urmele roller coaster-ului**

**drawCoasterHistory();**

**// Actualizăm timpul**

**t += dt;**

**}**

**// Funcție pentru desenarea unui traseu**

**function drawTrack(points) {**

**stroke(150); // Culorăm traseul în gri**

**strokeWeight(8); // Creștem grosimea liniei pentru a face drumul mai vizibil**

**noFill();**

**beginShape();**

**for (let i = 0; i < points.length; i++) {**

**let p = points[i];**

**vertex(p.x, p.y, p.z);**

**}**

**endShape();**

**}**

**// Funcție pentru simularea mișcării roller coaster-ului pe un traseu dat**

**function simulateCoaster(points) {**

**let pos = getCoasterPosition(points, t);**

**let x = pos.x;**

**let y = pos.y;**

**let z = pos.z;**

**let phi = pos.phi;**

**// Adăugăm poziția curentă la istoricul roller coaster-ului**

**coasterHistory.push(createVector(x, y, z));**

**// Limităm lungimea istoricului la 100 de poziții**

**if (coasterHistory.length > 100) {**

**coasterHistory.splice(0, 1);**

**}**

**// Desenăm roller coaster-ul la poziția curentă**

**push();**

**translate(x, y, z);**

**rotateZ(phi - HALF\_PI);**

**fill(0, 0, 255); // Colorăm roller coaster-ul în albastru**

**sphere(rollerCoasterRadius); // Roller coaster-ul ca sferă**

**pop();**

**}**

**// Funcție pentru a desena urmele roller coaster-ului**

**function drawCoasterHistory() {**

**noFill();**

**stroke(0, 0, 255); // Culorăm urmele în albastru**

**beginShape();**

**for (let i = 0; i < coasterHistory.length; i++) {**

**let pos = coasterHistory[i];**

**vertex(pos.x, pos.y, pos.z);**

**}**

**endShape();**

**}**

**// Funcție pentru a obține poziția și unghiul roller coaster-ului pe un anumit traseu la un anumit timp t**

**function getCoasterPosition(points, t) {**

**// Asigurăm că timpul rămâne între 0 și lungimea traseului**

**t = constrain(t, 0, points.length - 1);**

**let index = floor(t);**

**let p1 = points[index];**

**let p2 = points[index + 1];**

**let dx = p2.x - p1.x;**

**let dy = p2.y - p1.y;**

**let dz = p2.z - p1.z;**

**let phi = atan2(dy, dx);**

**let x = p1.x + dx \* (t - index);**

**let y = p1.y + dy \* (t - index);**

**let z = p1.z + dz \* (t - index);**

**return {**

**x: x,**

**y: y,**

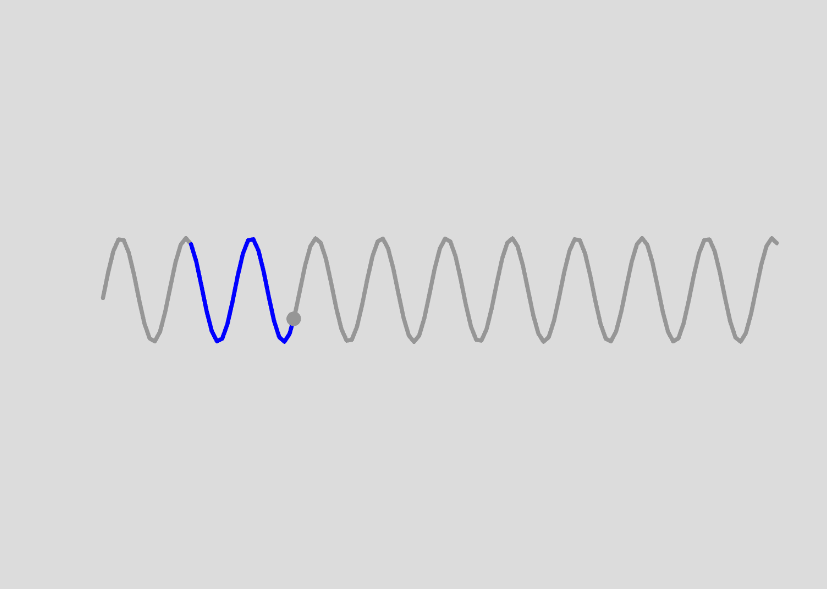
**z: z,**

**phi: phi**

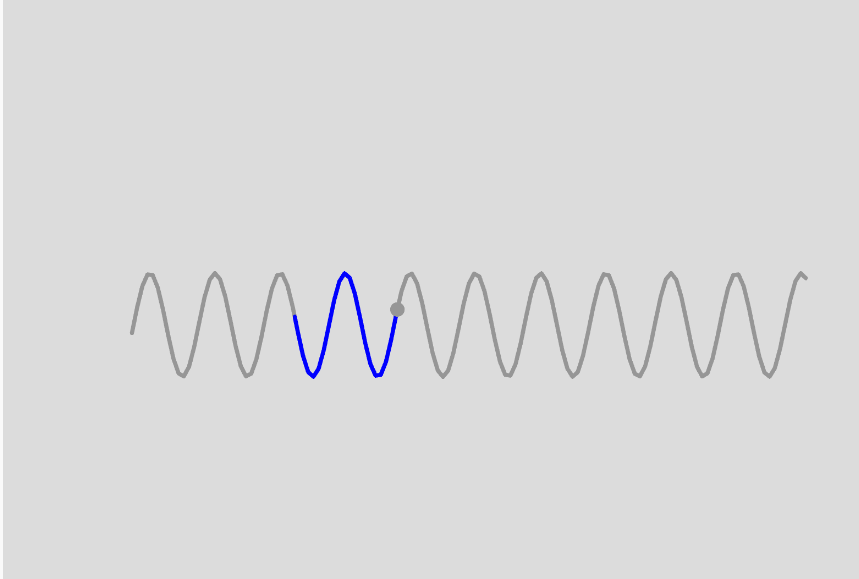
**};**

**}**

**Captura de ecran:**

****

**Fig.1**

****

**Fig.2**

**Concluzie:**

Prin dezvoltarea acestui program folosind biblioteca p5.js, am dobândit cunoștințe practice în modelarea proceselor 3D dinamice. Am învățat cum să folosesc funcțiile de translație și rotație pentru a poziționa și orienta obiecte în spațiu, precum și cum să creez și să gestionez un traseu tridimensional pentru simularea mișcării unui roller coaster. Această experiență este valoroasă în domeniile graficii computerizate, animației și simulării proceselor fizice, oferindu-ne posibilitatea de a crea și de a vizualiza diverse fenomene într-un mediu tridimensional interactiv. Este important să continuăm explorarea și dezvoltarea abilităților noastre în acest domeniu pentru a crea aplicații și simulări din ce în ce mai complexe și realiste.