Ministerul Educației și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatica

Raport

Lucrarea de laborator nr. 6

Grafică pe Calculator

Varianta 3

A efectuat:	Student grupa TI-231 FR	Apareci Aurica
A verificat:	asistent universitar	Ursu Adriana

Chișinău

2024

Cuprins

1. **Cadrul teoretic** 3
2. **Rezumat succint la temă** 4
3. **Listingul programului** 5
4. **Testarea aplicației** 6
5. **Concluzii** 8

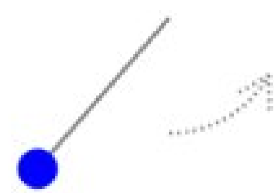
1. Cadrul teoretic

Tema: Modelarea proceselor 3D dinamice

Scopul lucrării: Obținerea cunoștințelor practice în modelarea proceselor 3D dinamice,

utilizând funcțiile standard de translație, și rotație din biblioteca p5.js.

Sarcina (conform variantei): 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.



Laborator nr. 6 Varianta 3

Pendul haotic

2. Rezumat succint la temă

Pentru realizarea simulării unui **pendul haotic** folosind biblioteca *p5.js*, au fost aplicate o serie de legi fizice fundamentale, care guvernează mișcarea pendulului și dinamica acestuia.

1. Legea mișcării pendulului

Pendulul se bazează pe principiile de bază ale mecanicii clasice, care descriu mișcarea oscilatorie. Formula de bază pentru accelerația unghiulară a pendulului este derivată din legile lui Newton pentru rotație: $\alpha = -\frac{g}{L}\sin(\theta)$ Unde:

α este accelerația unghiulară (rad/s²),

g este accelerația gravitațională (9.8 m/s²),

L este lungimea pendulului,

θ este unghiul pe care îl face pendulul cu poziția de echilibru (radiani).

2. Mișcarea oscilatorie și accelerația unghiulară

Formula de mai sus este aplicată pentru a calcula accelerația unghiulară a pendulului în funcție de poziția sa. AccelerațiA unghiulară este responsabilă de modificarea vitezei unghiulare a pendulului. În cod, accelerația unghiulară este actualizată pe baza gravitației, iar viteza unghiulară și unghiul pendulului sunt calculate iterativ:

```
let gravity = 0.4;
angleAcceleration = (-1 * gravity / length) * sin(angle);
```

3. Amortizarea mișcării (Damping)

În viața reală, mișcarea pendulului este încetinită de frecarea aerului și de alte forțe de rezistență. Această rezistență este cunoscută sub numele de amortizare și reduce treptat viteza pendulului. Pentru a modela acest comportament în simulare, am introdus un factor de amortizare care înmulțește viteza unghiulară a pendulului cu o valoare ușor mai mică decât 1 în fiecare iterație:

```
angleVelocity *= damping;
```

Mișcarea pendulului devine treptat mai lentă, simulând pierderea de energie din sistemul pendulului.

4. Haosul și perturbațiile

Pentru a simula un **pendul haotic**, a fost adăugată o mică forță externă variabilă, care perturbează mișcarea naturală a pendulului. Această forță este generată aleator și variază la fiecare iterație, simulând efectul unor factori externi care nu sunt constanți (precum vântul sau variații ale mediului).

Formula de calcul a accelerației unghiulare modificată pentru a include această forță externă este:

```
angleAcceleration += random(-forcePerturbation, forcePerturbation);
```

Aceasta introduce variații haotice în mișcare, făcând pendulul să oscileze neregulat, ducând la comportamente imprevizibile care sunt caracteristice sistemelor haotice.

5. Conservarea energiei mecanice

În absența forțelor de amortizare și a perturbațiilor, pendulul ar oscila la infinit, menținându-și energia mecanică totală. Energia mecanică totală este suma dintre energia cinetică și energia potențială gravitațională a pendulului:

Energia potențială este maximă atunci când pendulul este la extremitățile traiectoriei sale (când viteza unghiulară este zero și accelerația este maximă),

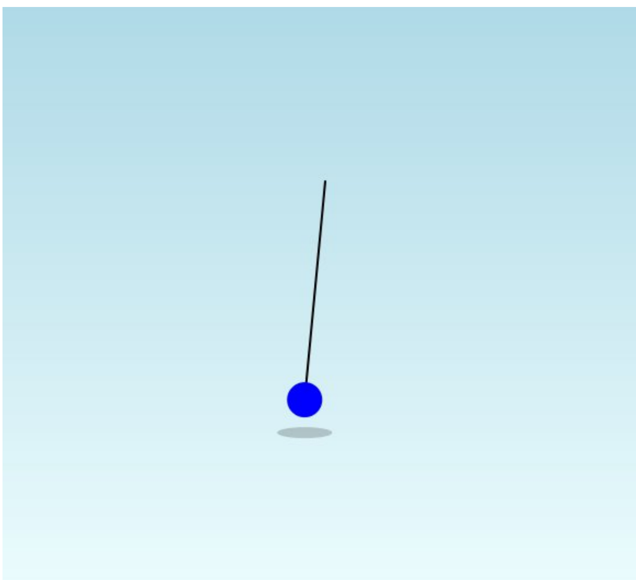
Energia cinetică este maximă atunci când pendulul trece prin poziția de echilibru (când viteza este maximă și accelerația este zero).

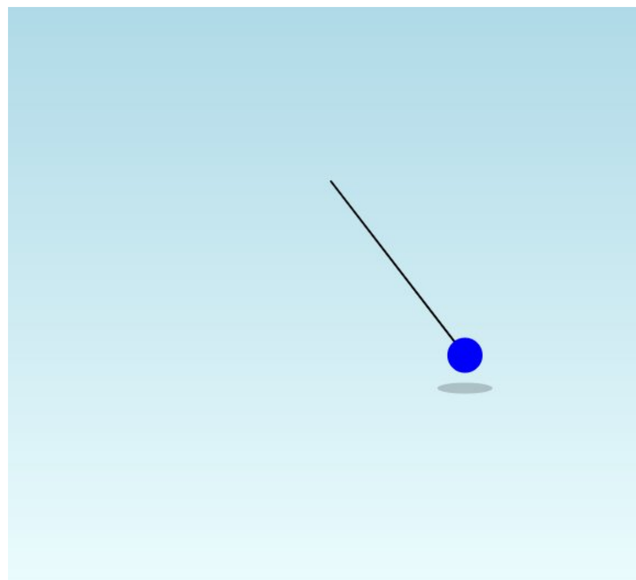
3. Listingul programului

```
let angle;
let angleVelocity;
let angleAcceleration;
let length;
let origin;
let bob;
let damping = 0.995;
let forcePerturbation;

function setup() {
  createCanvas(600, 600);
  origin = createVector(width / 2, 180);
  length = 200;
  angle = PI / 4;
  angleVelocity = 0.0;
  forcePerturbation = 0.01;
}

function draw() {
  setGradientBackground();

  let gravity = 0.4;
  angleAcceleration = (-1 * gravity / length) * sin(angle);
  angleAcceleration += random(-forcePerturbation, forcePerturbation);

  // Actualizează viteza și unghiul
  angleVelocity += angleAcceleration;
  angleVelocity *= damping; // Amortizare
  angle += angleVelocity;

  // Calculează poziția punctului final
  let bobX = length * sin(angle);
  let bobY = length * cos(angle);
  bob = createVector(bobX, bobY);
  bob.add(origin);

  stroke(0);
  strokeWeight(2);
  line(origin.x, origin.y, bob.x, bob.y);
  drawShadow(bob.x, bob.y);

  fill(0, 0, 255);
```

```
  ellipse(bob.x, bob.y, 32, 32);}
function setGradientBackground() {
  for (let y = 0; y < height; y++) {
    let inter = map(y, 0, height, 0, 1);
    let c = lerpColor(color(173, 216, 230), color(240, 255, 255), inter);
    stroke(c);
    line(0, y, width, y);
  }
}

function drawShadow(x, y) {
  noStroke();
  fill(0, 0, 0, 50);
  ellipse(x, y + 30, 50, 10);
}
```

4. Testarea aplicației





5. Concluzii

Simularea pendulului haotic realizată în p5.js integrează legile clasice ale fizicii, cum ar fi mișcarea oscilatorie și conservarea energiei, cu o componentă haotică prin forțe externe variabile. Calculăm accelerația unghiulară a pendulului, unde *g* reprezintă gravitația, *L* lungimea pendulului, iar *theta* unghiul de oscilație. Factorul de amortizare reduce treptat viteza unghiulară, simulând rezistența aerului. Componenta haotică este generată prin adăugarea unei forțe perturbatoare aleatoare, reflectând natura imprevizibilă a sistemelor fizice. Utilizăm funcții de desenare pentru a reprezenta pendulul și bob-ul, iar fundalul dinamic îmbunătățește vizualizarea. p5.js permite extinderea ușoară a simulării, adăugând interactivitate și caracteristici noi. Această simulare ilustrează complexitatea comportamentului haotic, evidențiind puterea p5.js ca instrument de învățare în fizică și programare.