

# MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA

**Universitatea Tehnică a Moldovei**

# Facultatea Calculatoare, Informatică şi Microelectronică Departamentul Inginerie Software și Automatică

**Cristian Brinza FAF-212**

Report

*Laboratory work n.4*

***of Computer Graphics***

Checked by:

**Olga Grosu,** *university assistant*

DISA, FCIM, UTM

# Chișinău – 2022

1. **The purpose of the laboratory work (formulated by the student according to the problem to be solved);**

Learning and exercising the GC concepts, understanding and using them to create laboratories work which represents and simulates the use in the field.

# The condition / conditions of the problems:

Analyze the Example from Introduction and the Example from Chapter 1. Vectors and make the Exercise 1.1-1.8.

*The task above you must do before the periodic assessment no. 1*

*The task below is for the week after assessment.*

A: Do the sketch using the function:

• randomGaussian()

• randomSeed()

• random()

• noiseDetail()

• noiseSeed()

• noise()

• map()

B: Vectors project:

Develop a set of rules for simulating the real-world behavior of a creature, such as a nervous fly, swimming fish, hopping bunny, slithering snake, etc. Can you control the object’s motion by only manipulating the acceleration? Try to give the creature a personality through its behavior (rather than through its visual design).

# The program code, having relevant comments in it:

**Exercise 1.1**

Find something you’ve previously made in Processing using separate x and y variables and use PVectors instead.

//https://editor.p5js.org/

let letter;

function setup() {

  createCanvas(600, 600);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.move();

  letter.bounce();

}

class Letter {

  // create the instance of the object

  constructor() {

    this.pos = createVector(300, 300);

    this.vel = createVector(1, 2);

  }

  show() {

    textSize(70);

    text("Z", this.pos.x, this.pos.y);

  }

  move() {

    this.pos.add(this.vel);

  }

  bounce() {

    if (this.pos.x > width-40 || this.pos.x < 0) {

      this.vel.x = -this.vel.x;

    }

    if (this.pos.y > width || this.pos.y-50 < 0) {

      this.vel.y = -this.vel.y;

    }

  }

}

**Exercise 1.2**

Take one of the walker examples from the introduction and convert it to use PVectors.

//https://editor.p5js.org/

let letter;

function setup() {

  createCanvas(400, 400);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.step();

}

class Letter {

  constructor() {

    this.pos = createVector(200, 200);

    this.t = createVector(0, 10000);

  }

  show() {

    textSize(70);

    text("Z", this.pos.x, this.pos.y);

  }

  step() {

    this.pos.x = map(noise(this.t.x), 0, 1, 0, width);

    this.pos.y = map(noise(this.t.y), 0, 1, 0, height);

    this.t.x += 0.01;

    this.t.y += 0.01;

  }

}

**Exercise 1.3**

Extend the bouncing ball with vectors example into 3D. Can you get a sphere to bounce around a box?

//https://editor.p5js.org/

let ball;

function setup() {

  createCanvas(600, 600, WEBGL);

  ball = new Ball();

}

function draw() {

  background(180);

  fill(24, 50);

  push();

  ball.show();

  ball.move();

  ball.bounce();

  pop();

  push();

  translate(0, 0, -width / 2);

  box(width, height, width)

  pop();

}

class Ball {

  constructor() {

    this.pos = createVector(0, 0, 0);

    this.vel = createVector(4, 7, 5);

    this.size = 40; // radius

  }

  show() {

    translate(this.pos.x, this.pos.y, this.pos.z)

    sphere(this.size);

  }

  move() {

    this.pos.add(this.vel);

  }

  bounce() {

    this.vel.x = bounce(this.pos.x, this.vel.x, this.size);

    this.vel.y = bounce(this.pos.y, this.vel.y, this.size);

    this.vel.z = bounce(this.pos.z, this.vel.z, this.size);

     if (this.pos.x > width/2 || this.pos.x < -height/2) {

       this.vel.x = -this.vel.x;

     }

     if (this.pos.y > width/2 || this.pos.y < -height/2) {

       this.vel.y = -this.vel.y;

     }

     if (this.pos.z > width/2 || this.pos.z < -height/2) {

       this.vel.z = -this.vel.z;

     }

  }

}

function bounce(pos, vel, size) {

  if (pos > (width / 2 - (size\*2)) || pos < -(height / 2 - (size\*2))) {

    vel = -vel;

  }

  return vel;

}

**Exercise 1.4**

Write the limit() function for the PVector class

void limit(float max) {

    if (magSq() > max\*max) {

      normalize();

      mult(max);

    }

  }

**Exercise 1.5**

Create a simulation of a car (or runner) that accelerates when you press the up key and brakes when you press the down key.

//https://editor.p5js.org/

let letter;

function setup() {

  createCanvas(600, 600);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.move();

  // letter.bounce();

  letter.checkEdges();

  // gas and break pedals!

  if (keyIsPressed == true) {

    if (keyCode == UP\_ARROW) {

      letter.accelerate();

    } else if (keyCode == DOWN\_ARROW) {

      letter.break();

    }

  }

}

class Letter {

  // create the instance of the object

  // how do you want to set it up?

  constructor() {

    this.pos = createVector(300, 300);

    // start in a resting position

    this.vel = createVector(0, 0);

    this.acc = createVector(0.001, 0);

    this.topspeed = 100;

  }

  show() {

    textSize(24);

    text("O", this.pos.x, this.pos.y);

  }

  // control motion entirely with acceleration

  move() {

    this.vel.add(this.acc);

    this.vel.limit(this.topspeed);

    this.pos.add(this.vel);

  }

   bounce() {

    if (this.pos.x > width || this.pos.x < 0) {

      this.vel.x = -this.vel.x;

    }

    if (this.pos.y > width || this.pos.y < 0) {

      this.vel.y = -this.vel.y;

    }

  }

  checkEdges() {

    if (this.pos.x > width) this.pos.x = 0;

    else if (this.pos.x < 0) this.pos.x = width

    if (this.pos.y > height) this.pos.y = 0;

    else if (this.pos.y < 0) this.pos.y = height;

  }

  accelerate() {

    this.topspeed = 10;

    this.acc.mult(1.2);

  }

  break () {

    this.topspeed = 0;

    // and reset acc vals

    this.acc.x = 0.001;

    this.acc.y = 0;

    // this.vel.mult(0); // I can't tell a difference using this

  }

}

**Exercise 1.6**

Referring back to the [Introduction](https://natureofcode.com/book/chapter-1-vectors/#intro_section6), implement acceleration according to Perlin noise.

//https://editor.p5js.org/

let letter;

function setup() {

  createCanvas(600, 600);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.move();

  letter.checkEdges();

  // gas and break pedals!

  if (keyIsPressed == true) {

    if (keyCode == UP\_ARROW) {

      letter.accelerate();

    } else if (keyCode == DOWN\_ARROW) {

      letter.break(0);

    }

  }

}

class Letter {

  // create the instance of the object

  constructor() {

    this.pos = createVector(300, 300);

    // start in a resting position

    this.vel = createVector(0, 0);

    // this.acc = createVector(-0.001, 0.01);

     //this.acc = createVector(noise(random(width)), noise(random(height)));

    this.topspeed = 5;

    this.time = createVector(0, 10000);

  }

  show() {

    textSize(24);

    text("0", this.pos.x, this.pos.y);

  }

  // control motion entirely with acceleration

  move() {

    let x = noise(this.time.x);

    let y = noise(this.time.y);

    this.acc = createVector(x, y)

    this.vel.add(this.acc);

    this.vel.limit(this.topspeed);

    this.pos.add(this.vel);

    this.time.add(0.01);

  }

  checkEdges() {

    if (this.pos.x > width) {

      this.pos.x = 0;

    } else if (this.pos.x < 0) {

      this.pos.x = width;

    }

    if (this.pos.y > height) {

      this.pos.y = 0;

    } else if (this.pos.y < 0) {

      this.pos.y = height;

    }

  }

  accelerate() {

    this.topspeed = 100;

    this.acc.mult(0.5);

  }

  break() {

    this.topspeed = 0;

     this.acc.x = -0.001;

     this.acc.y = 0.01;

  }

}

**Exercise 1.7**

Translate the following pseudocode to code using static or non-static functions where appropriate.

* The PVector v equals (1,5).
* The PVector u equals v multiplied by 2.
* The PVector w equals v minus u.
* Divide the PVector w by 3.

PVector v = new PVector(1,5);

PVector u = PVector.mult(v,5);

PVector w = PVector.sub(v,u);

w.div(3);

**Exercise 1.8**

Try implementing the above example with a variable magnitude of acceleration, stronger when it is either closer or farther away.

//https://editor.p5js.org/

let letter;

function setup() {

  createCanvas(600, 600);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.move();

}

class Letter {

  constructor() {

    this.pos = createVector(300, 300);

    this.vel = createVector(0, 0); // resting position

    this.acc = createVector(0, 0);

    this.topspeed = 10;

    this.mag = 0.5;

  }

  show() {

    textSize(24);

    text("O", this.pos.x, this.pos.y);

  }

  // control motion entirely with acceleration

  move() {

    let mouse = createVector(mouseX, mouseY);

    // 1. Compute direction

    let dir = p5.Vector.sub(mouse, this.pos);

    // create an arbitrary distance

    let num = createVector(20, 20);

    // Farther away: If the diff is more

    // decrease the mag

    if (dir > num) this.mag -= 0.01

    else this.mag += 0.01

    // 2. Set magnitude (normalize and scale);

     dir.normalize();

     dir.mult(0.9);

    dir.setMag(this.mag);

    // 3. Accelerate

    this.acc = dir;

    // 4. Control velocity with acceleration

    this.vel.add(this.acc);

    this.vel.limit(this.topspeed);

    this.pos.add(this.vel);

  }

}

**LAB 4a**

randomGaussian()

//https://editor.p5js.org/

let distribution = new Array(360);

function setup() {

   createCanvas(600, 600);

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

     distribution[i] = floor(randomGaussian(0, 55));

   }

 }

function draw() {

   background(204);

  translate(width / 2, width / 2);

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

     rotate(TWO\_PI / distribution.length);

     stroke(0);

     let dist = abs(distribution[i]);

     line(0, 0, dist, 0);

   }

  describe(`black lines radiate from center of canvas.

     The size changes each render.`);

 }

randomSeed()

//https://editor.p5js.org/

function setup() {

    createCanvas(400, 400);

    background(30);

    fill(255, 0, 0);

    noStroke();

    // Pick a random seed

    var seed = random(10000);

    // Log it so you can see the number

      console.log(seed);

    // Use the seed in the random function.

    // If you like the result, replace "seed" with the number

    // to lock your sketch to that

      randomSeed(seed);

    // Draw five random ellipses

    for(var i = 0; i < 5; i++) {

         ellipse(random(width), random(height), 40, 40);

    }

  }

random()

//https://editor.p5js.org/

function setup() {

    createCanvas(710, 400);

    background(0);

    strokeWeight(20);

    frameRate(2);

  }

  function draw() {

    for (let i = 0; i < width; i++) {

      let r = random(255);

      stroke(r);

      line(i, 0, i, height);

    }

  }

noiseDetail()

//https://editor.p5js.org/

let noiseVal;

 let noiseScale = 0.02;

function setup() {

   createCanvas(100, 100);

 }

function draw() {

   background(0);

   for (let y = 0; y < height; y++) {

     for (let x = 0; x < width / 2; x++) {

       noiseDetail(2, 0.2);

       noiseVal = noise((mouseX + x) \* noiseScale, (mouseY + y) \* noiseScale);

       stroke(noiseVal \* 255);

       point(x, y);

       noiseDetail(8, 0.65);

       noiseVal = noise(

         (mouseX + x + width / 2) \* noiseScale,

         (mouseY + y) \* noiseScale

       );

       stroke(noiseVal \* 255);

       point(x + width / 2, y);

     }

   }

   describe(`2 vertical grey smokey patterns affected by

     mouse x-position and noise.`);

 }

noiseSeed()

//https://editor.p5js.org/

let xoff = 0.0;

function setup() {

  noiseSeed(99);

  stroke(0, 10);

}

function draw() {

  xoff = xoff + .01;

  let n = noise(xoff) \* width;

  line(n, 0, n, height);

  describe(`vertical grey lines drawing in pattern affected by noise.`);

}

noise()

//https://editor.p5js.org/

let xoff = 0.0;

function draw() {

  background(204);

  xoff = xoff + 0.01;

  let n = noise(xoff) \* width;

  line(n, 0, n, height);

  describe(`vertical line moves left to right with updating

    noise values.`);

}

map()

//https://editor.p5js.org/

function setup() {

    noStroke();

  }

  function draw() {

    background(204);

    let x1 = map(mouseX, 0, width, 25, 75);

    ellipse(x1, 25, 25, 25);

    //This ellipse is constrained to the 0-100 range

    //after setting withinBounds to true

    let x2 = map(mouseX, 0, width, 0, 100, true);

    ellipse(x2, 75, 25, 25);

    describe(`Two 25×25 white ellipses move with mouse x.

      Bottom has more range from X`);

  }

**LAB 4b**

//https://editor.p5js.org/

let letter;

// let pdir;

function setup() {

  createCanvas(800, 800);

  letter = new Letter();

}

function draw() {

  background(220);

  letter.show();

  letter.move();

}

class Letter {

  constructor() {

    this.pos = createVector(800, 800);

    this.vel = createVector(0, 0); // resting position

    this.acc = createVector(0, 0);

    this.topspeed = 10;

    this.mag = 0.5;

  }

  show() {

    textSize(24);

    text("CAT", this.pos.x, this.pos.y);

  }

  // control motion entirely with acceleration

  move() {

    let mouse = createVector(mouseX, mouseY);

    // 1. Compute direction

    let dir = p5.Vector.sub(mouse, this.pos);

    // create an arbitrary distance

    let num = createVector(200, 200);

    // Farther away: If the diff is more

    // decrease the mag

    if (dir > num) this.mag -= 0.01

    else this.mag += 0.01

    // Closer: If diff is less,

    // decrease the mag

    // goodbye!

    if (dir > num) this.mag += 0.008

     else this.mag -= 0.008

    // 2. Set magnitude (normalize and scale);

    // dir.normalize();

    // dir.mult(0.5);

    dir.setMag(this.mag);

    // 3. Accelerate

    this.acc = dir;

    // 4. Control velocity with acceleration

    this.vel.add(this.acc);

    this.vel.limit(this.topspeed);

    this.pos.add(this.vel);

  }

}

1. **Screen printing of program execution;**

# The student's conclusions regarding the content of the laboratory work with personal reflections on what was achieved; difficulties encountered and how he/she got over them (if he/she got over them). Where did he find the answer? (specify the links to sources that help you to get the answer).

The laboratory work had a big impact in the quality of my development as a future Software Engineer in Computer Graphics. I enjoyed the process, developed my creativity and codding skills, understood better the IDE and the code functionality.

Biography: <http://learningprocessing.com/examples/>

<https://codebeautify.org/javaviewer>

<https://else.fcim.utm.md/course/view.php?id=573>