

**MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA**

**Universitatea Tehnică a Moldovei**

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

**Gusev Roman FAF-222**

Report

*Laboratory work n.4*

*Point B*

***of Computer Graphics***

Checked by:

**Olga Grosu,** *university assistant*

DISA, FCIM, UTM

**Chișinău – 2023**

**1. Purpose of the Laboratory Work**

In point B of laboratory Work nr. 4, I had to draw an ecosystem, that will help me understand a real project better and how to use all of the previous laboratory works

**2. Condition of the Laboratory Work**

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).

**3. The program code**

**Main Class:**

ArrayList<Insect> insects = new ArrayList<Insect>();

ArrayList<Food> foods = new ArrayList<Food>();

int numInsects = 5;

void setup() {

size(1000, 1000);

for (int i = 0; i < numInsects; i++) {

insects.add(new Insect());

}

}

void draw() {

background(220);

// Create new food occasionally

if (random(1) < 0.2) {

foods.add(new Food());

}

// Update and display food

for (Food food : foods) {

food.display();

}

// Update and display insects

for (int i = insects.size() - 1; i >= 0; i--) {

Insect Insect = insects.get(i);

Insect.update();

Insect.display();

// Check for collision with food

for (int j = foods.size() - 1; j >= 0; j--) {

Food food = foods.get(j);

if (Insect.position.dist(food.position) < 10) {

foods.remove(j); // Remove the food

Insect.eat();

}

}

// Check if the Insect has reached maximum size

if (Insect.isMaxSize()) {

insects.remove(i); // Remove the Insect

}

// Check for the nearest food and move towards it

if (!foods.isEmpty()) {

Food nearestFood = findNearestFood(Insect.position);

Insect.moveTo(nearestFood.position);

}

}

// Randomly create new flies

if (random(1) < 0.015) { // Adjust the probability for new fly appearance

insects.add(new Insect());

}

textSize(25);

fill(0);

text("Number of Insects: " + insects.size(), 50, 50);

}

Food findNearestFood(PVector InsectPosition) {

Food nearestFood = null;

float minDistance = Float.MAX\_VALUE;

for (Food food : foods) {

float distance = InsectPosition.dist(food.position);

if (distance < minDistance) {

minDistance = distance;

nearestFood = food;

}

}

return nearestFood;

}

**Food Class:**

class Food {

PVector position;

Food() {

position = new PVector(random(width), random(height));

}

void display() {

// Display the food

fill(0, 255, 0);

noStroke();

ellipse(position.x, position.y, 10, 10);

}

}

**Insect Class:**

class Insect {

PVector position;

PVector velocity;

PVector acceleration;

float maxSpeed = 3;

float maxForce = 0.1;

float maxSize = 25;

float size = 5;

Insect() {

position = new PVector(random(width), random(height));

velocity = PVector.random2D();

acceleration = new PVector(0, 0);

}

void update() {

// Apply jittery behavior

applyJitter();

// Update position, velocity, and acceleration

velocity.add(acceleration);

velocity.limit(maxSpeed);

position.add(velocity);

// Reset acceleration

acceleration.mult(0);

// Keep the Insect within the canvas

if (position.x > width) position.x = 0;

if (position.x < 0) position.x = width;

if (position.y > height) position.y = 0;

if (position.y < 0) position.y = height;

}

void applyJitter() {

// Apply random acceleration to simulate nervousness

PVector jitter = PVector.random2D();

jitter.mult(0.2); // Adjust jitteriness here

acceleration.add(jitter);

// Steer away from the mouse cursor

PVector desired = PVector.sub(position, new PVector(mouseX, mouseY));

float d = desired.mag();

if (d < 100) {

desired.setMag(map(d, 0, 100, 0, maxSpeed));

PVector steer = PVector.sub(desired, velocity);

steer.limit(maxForce);

acceleration.add(steer);

}

}

void display() {

// Display the Insect with wings

stroke(0);

fill(127, 127, 0);

ellipse(position.x, position.y, size, size);

// Draw wings

fill(255, 255, 0);

float wingLength = 5;

float wingWidth = 2;

pushMatrix();

translate(position.x, position.y);

rotate(velocity.heading());

rect(-wingLength, -wingWidth, wingLength \* 2, wingWidth \* 2);

popMatrix();

}

void eat() {

size += 1;

}

boolean isMaxSize() {

return size >= maxSize;

}

void moveTo(PVector target) {

// Move towards the target

PVector desired = PVector.sub(target, position);

float d = desired.mag();

if (d > 0) {

desired.setMag(maxSpeed);

PVector steer = PVector.sub(desired, velocity);

steer.limit(maxForce);

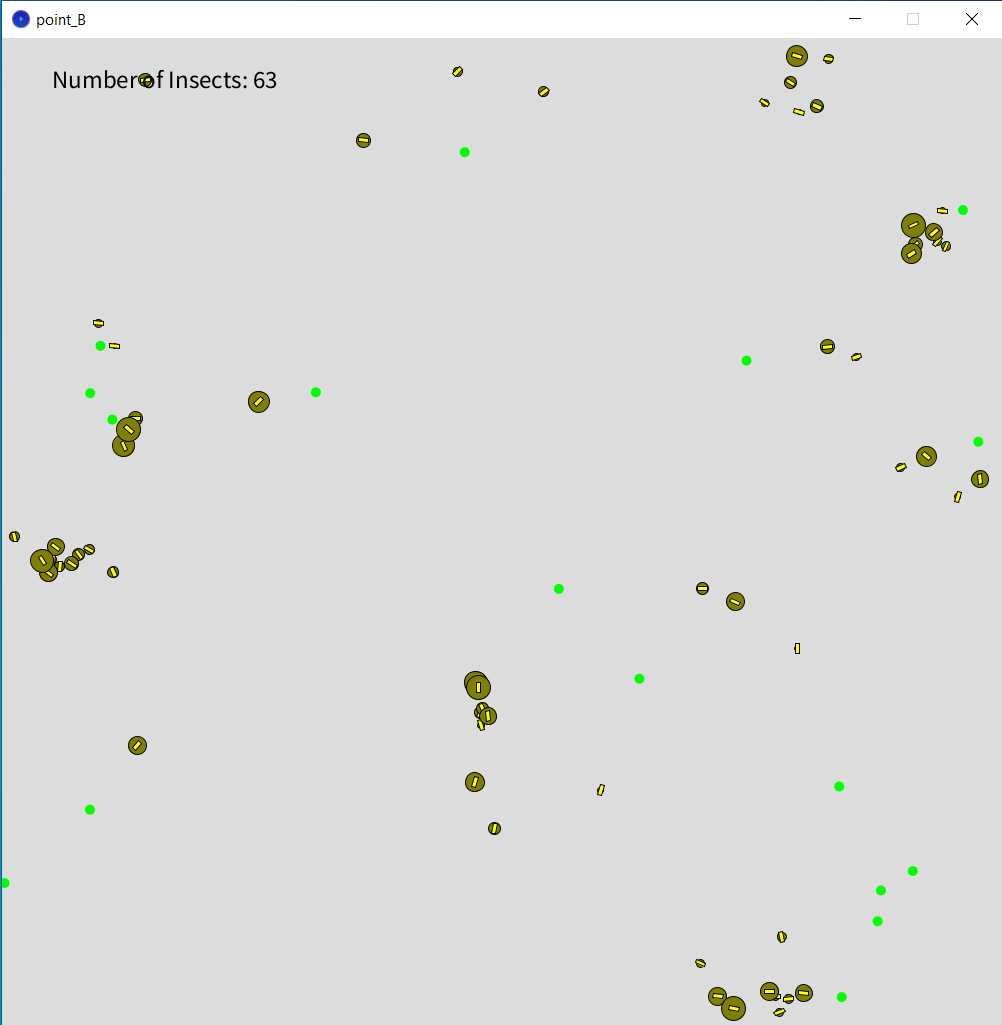
acceleration.add(steer);

}

}

}

**4. Screen printing of program execution**

****

**5. Conclusion**

By the end of this Laboratory Work nr. 4 Point B, I accomplished given task exactly with the requirements. I created a canvas where appears insects, with a random rate, appears food, same with random rate. Insects are moving randomly when there is no food on the canvas, if food is present, they are moving towards the food, and when they collide, insect eats the food and grows in size. At a certain size, they die. I simulated an ecosystem of insects. In conclusion I want to say that this project was very fun to do, because it was close to a real situation. I can say that the laboratory work was accomplished with success.