Selected files

3 printable files

```
Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Box.java
Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Main.java
Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Particle.java
```

Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Box.java

```java
import java.util.ArrayList;
import java.util.List;
import java.util.Random;

class Box {
    private static Box instance;
    private int width;
    private int height;
    private List<Particle> particles;

    private Box(int width, int height) {
        this.width = width;
        this.height = height;
        particles = new ArrayList<>();
        Random random = new Random();
        for (int i = 0; i < 3; i++) {
            particles.add(new Particle(random.nextInt(width), random.nextInt(height)));
        }
    }

    public static Box getInstance() {
        if (instance == null) {
            instance = new Box(10, 10); // Default size of the box
        }
        return instance;
    }

    public void simulate(int steps) {
        for (int i = 0; i < steps; i++) {
            moveParticles();
            checkCollisions();
            System.out.println("Step " + (i + 1) + ": Number of particles = " + particles.size());
            visualizeBox();
        }
    }

    private void moveParticles() {
        for (Particle particle : particles) {
            particle.moveRandomly();
        }
    }

    private void checkCollisions() {
        List<Particle> newParticles = new ArrayList<>();
        for (Particle particle : particles) {
            for (Particle other : particles) {
                if (particle != other && particle.getX() == other.getX() && particle.getY() == other.getY()) {
                    newParticles.add(new Particle(particle.getX(), particle.getY()));
                }
            }
        }
        particles.addAll(newParticles);
    }

    private void visualizeBox() {
        for (int i = 0; i < height; i++) {
            for (int j = 0; j < width; j++) {
                boolean isParticle = false;
                for (Particle particle : particles) {
                    if (particle.getX() == j && particle.getY() == i) {
                        isParticle = true;
                        break;
                    }
                }
                if (isParticle) {
                    System.out.print("* ");
                } else {
                    System.out.print("- ");
                }
            }
            System.out.println();
        }
        System.out.println("-------------------------");
    }

    public int getWidth() {
        return width;
    }

    public int getHeight() {
        return height;
    }
}

```

Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Main.java

```java
public class Main {
    public static void main(String[] args) {
        Box box = Box.getInstance();
        box.simulate(5); // Simulate movement for 5 steps
    }
}
```

Lab 2\Question 4 (Bonus) Particle Behavior in the Box Simulation\Particle.java

```java

import java.util.Random;

// Enum for directions
enum Direction {
    NORTH, NORTHEAST, EAST, SOUTHEAST, SOUTH, SOUTHWEST, WEST, NORTHWEST
}

// Particle class
class Particle {
    private int x;
    private int y;

    public Particle(int x, int y) {
        this.x = x;
        this.y = y;
    }

    public void moveRandomly() {
        Random random = new Random();
        Direction direction = Direction.values()[random.nextInt(Direction.values().length)];

        switch (direction) {
            case NORTH:
                if (y > 0)
                    y--;
                break;
            case NORTHEAST:
                if (y > 0 && x < Box.getInstance().getWidth() - 1) {
                    y--;
                    x++;
                }
                break;
            case EAST:
                if (x < Box.getInstance().getWidth() - 1)
                    x++;
                break;
            case SOUTHEAST:
                if (y < Box.getInstance().getHeight() - 1 && x < Box.getInstance().getWidth() - 1) {
                    y++;
                    x++;
                }
                break;
            case SOUTH:
                if (y < Box.getInstance().getHeight() - 1)
                    y++;
                break;
            case SOUTHWEST:
                if (y < Box.getInstance().getHeight() - 1 && x > 0) {
                    y++;
                    x--;
                }
                break;
            case WEST:
                if (x > 0)
                    x--;
                break;
            case NORTHWEST:
                if (y > 0 && x > 0) {
                    y--;
                    x--;
                }
                break;
        }
    }

    public int getX() {
        return x;
    }

    public int getY() {
        return y;
    }
}

```