JAVA PROGRAMMING PROJECT

Group 4

Presentation Title: 2D Snake Game

Prof. Gurleen Kaur

Georgian College

Presented by:

1. Amninder Kaur
2. Khushdeep Kaur
3. Deepak Kumar
4. Ramandeep Kaur

Due Date: 31 March 2023

INTRODUCTION

Good day everyone, we are Group 4 and we are excited to present our project on the Snake game developed with the help of Java. Our group consists of Amninder Kaur, Khushdeep Kaur, Deepak Kumar, Ramandeep Kaur and we have worked together to create a fun and challenging game for our audience.

For those who are not familiar, the Snake game is a classic video game that has been around for decades. It was first introduced in the late 1970s and gained popularity in the 1990s, thanks to its simple yet addictive gameplay. The objective of the game is to control a snake and eat as much food as possible without colliding with the walls or the snake's own body.

Our group decided to take on the challenge of developing the Snake game using Java, one of the most popular programming languages today. We also wanted to learn more about game development and programmingin general, and we felt that this project was a great opportunity to do so.

Throughout the development process, we faced several challenges and setbacks, but we persevered and overcame them with the help of each other and our instructor. We are proud to say that we have created a fully functional and enjoyable Snake game that we believe will provide hours of entertainment for our audience.

In the rest of our presentation, we will showcase the features and mechanics of our game, discuss the technologies and programming concepts we used, and share our overall experience and learnings from this project. We hope that you will enjoy our presentation and have fun playing our game. Thank you for your attention.

Contributions:

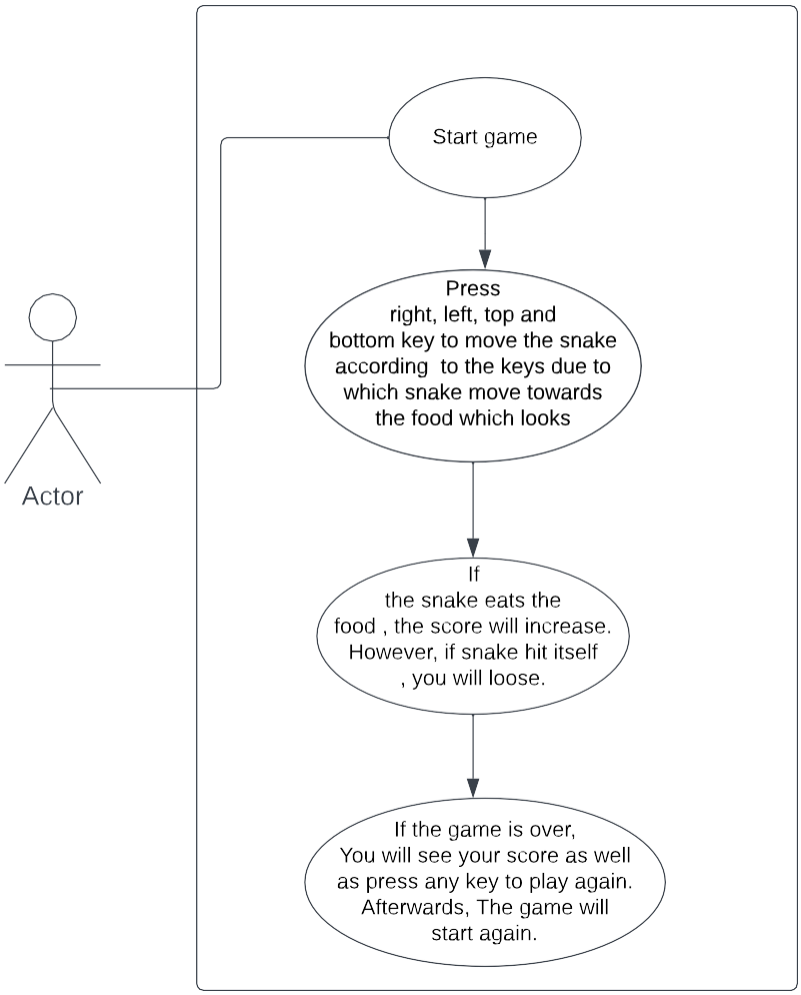
1. **Amninder Kaur**: Handling the class of graphics.
2. **Khushdeep Kaur**: Handling the class of food.
3. **Deepak Kumar**: Handling the class of game and main class.
4. **Ramandeep Kaur**: Not Responding.

The use case diagram, flow chart as well as pdf is prepared on zoom by everyone's contribution.

We are using the following libraries in our project:

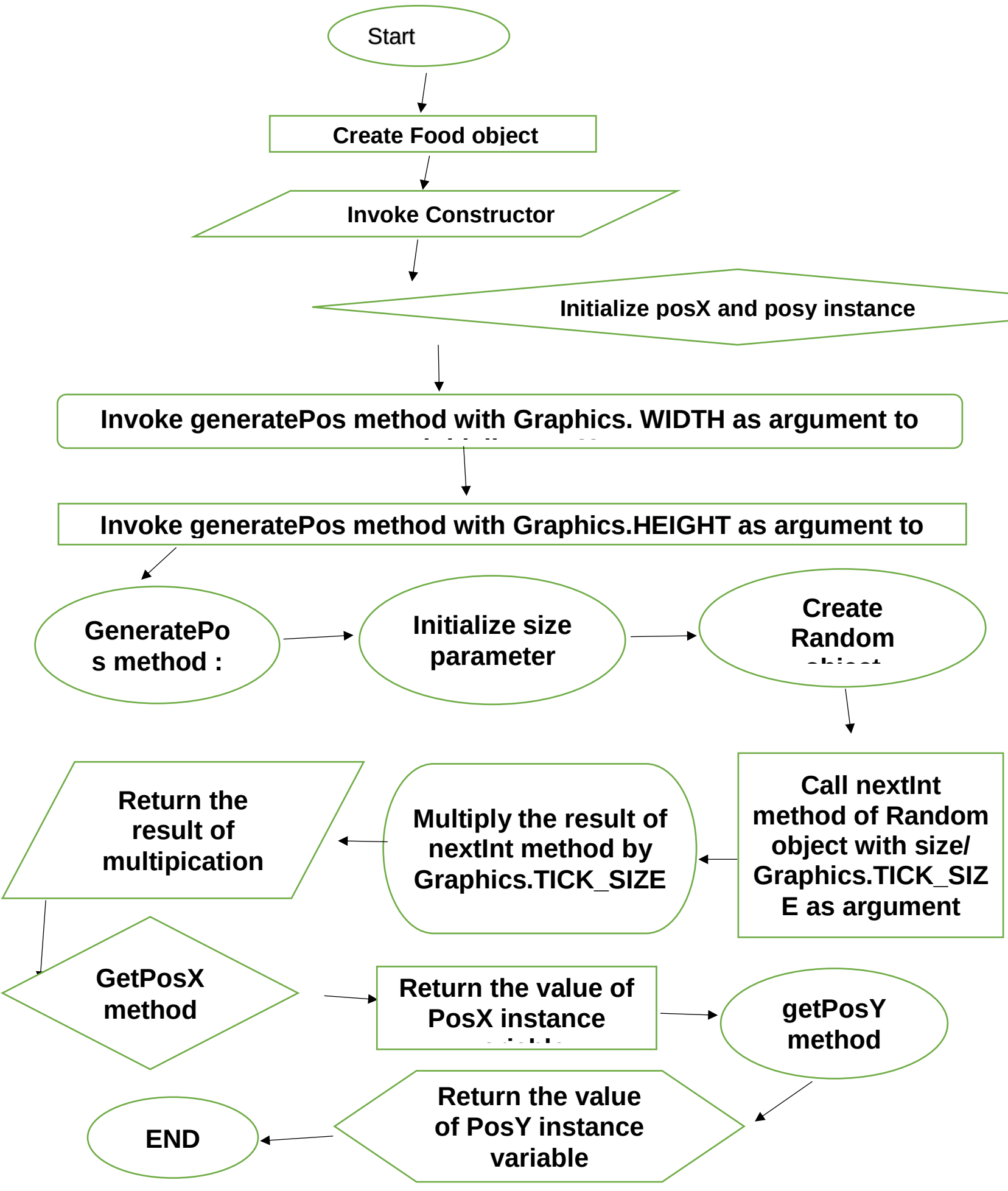
1. import java.util.Random
2. import java.swing.*
3. import java.awt.*
4. import java.awt.event.ActionEvent
5. import java.awt.event.ActionListener
6. import java.awt.event.KeyAdapter
7. import java.awt.event.KeyEvent

USE CASE DIAGRAM

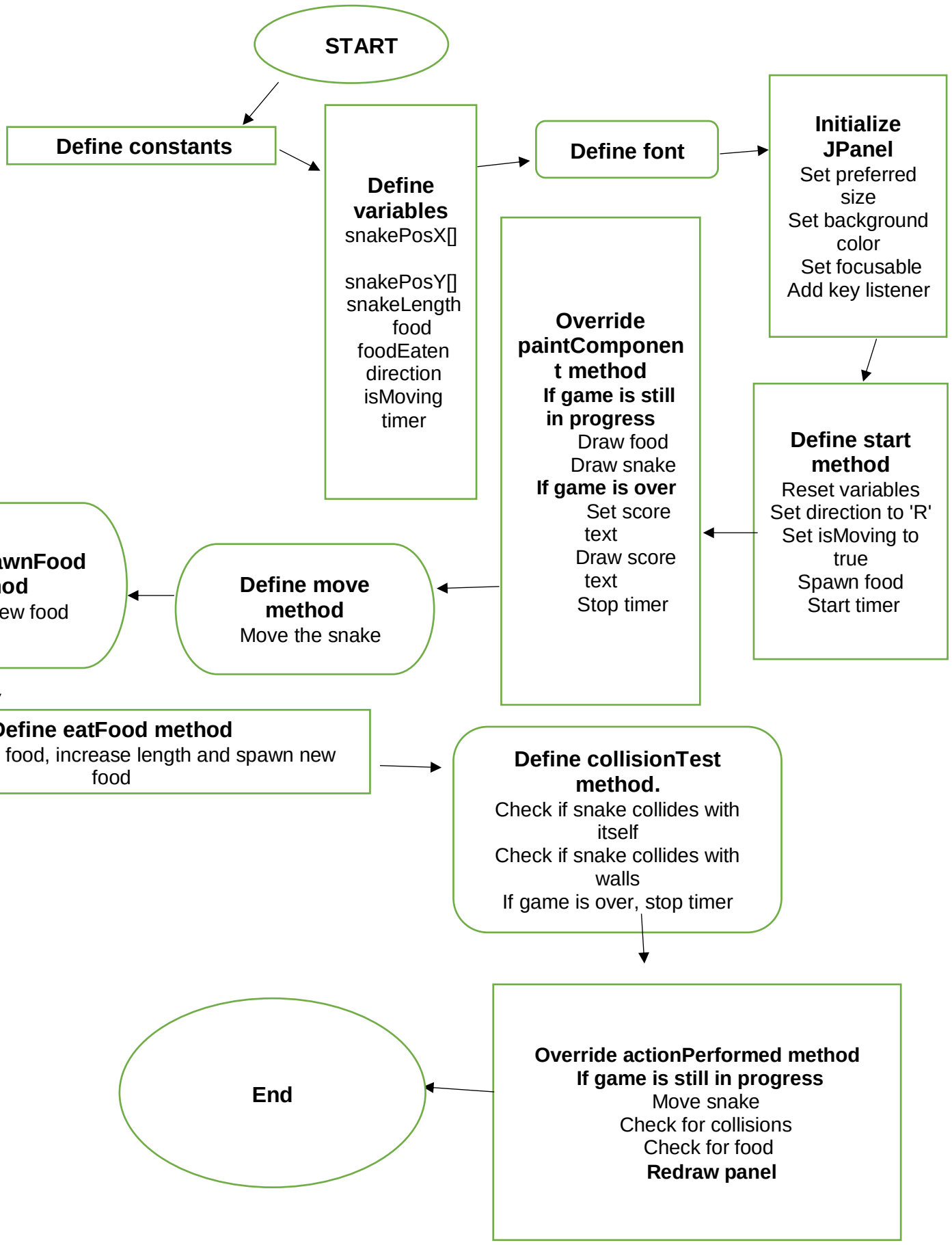


FLOWCHART OF ALL CLASSES:

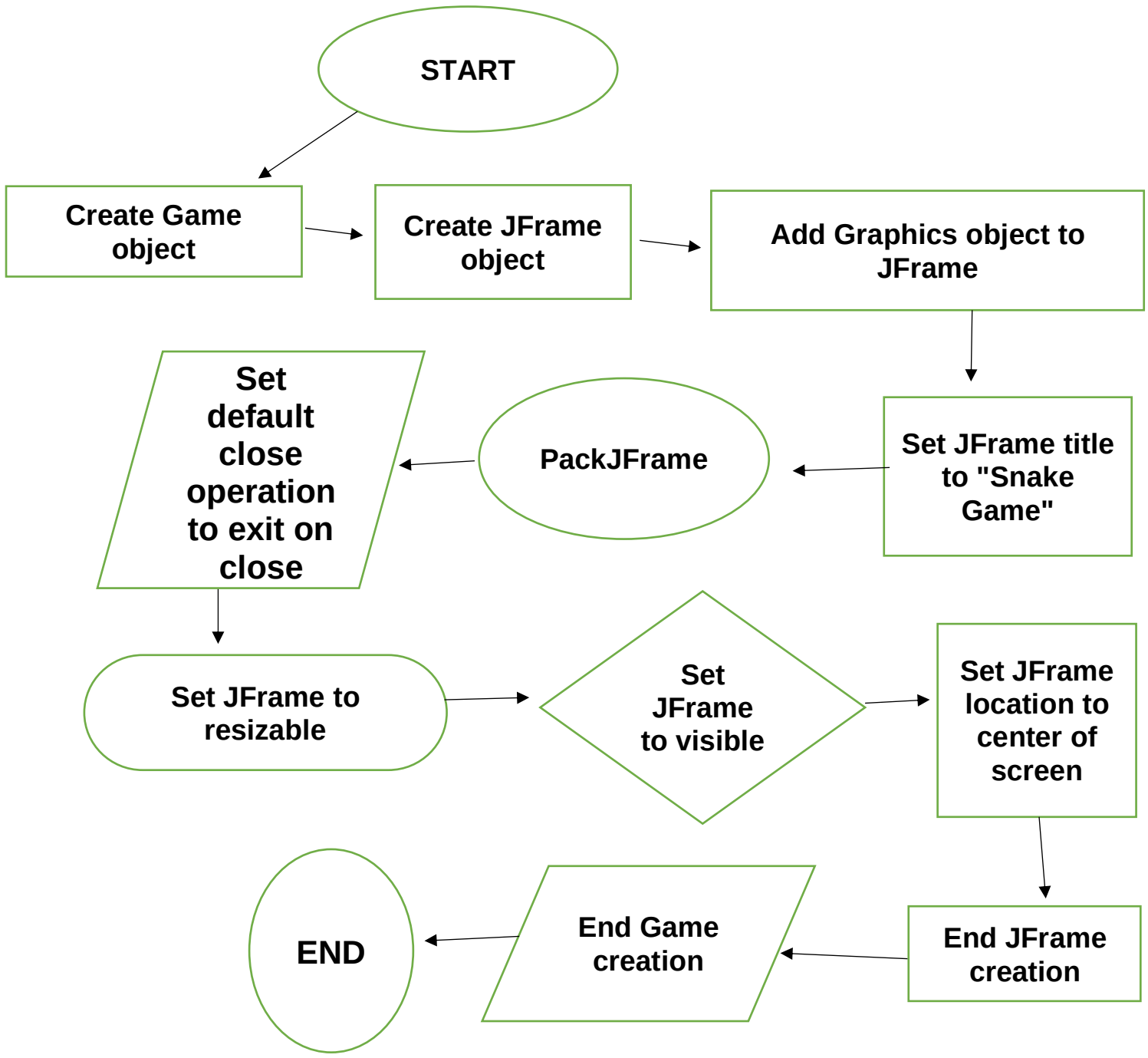
(1) FLOWCHART OF FOOD CLASS:



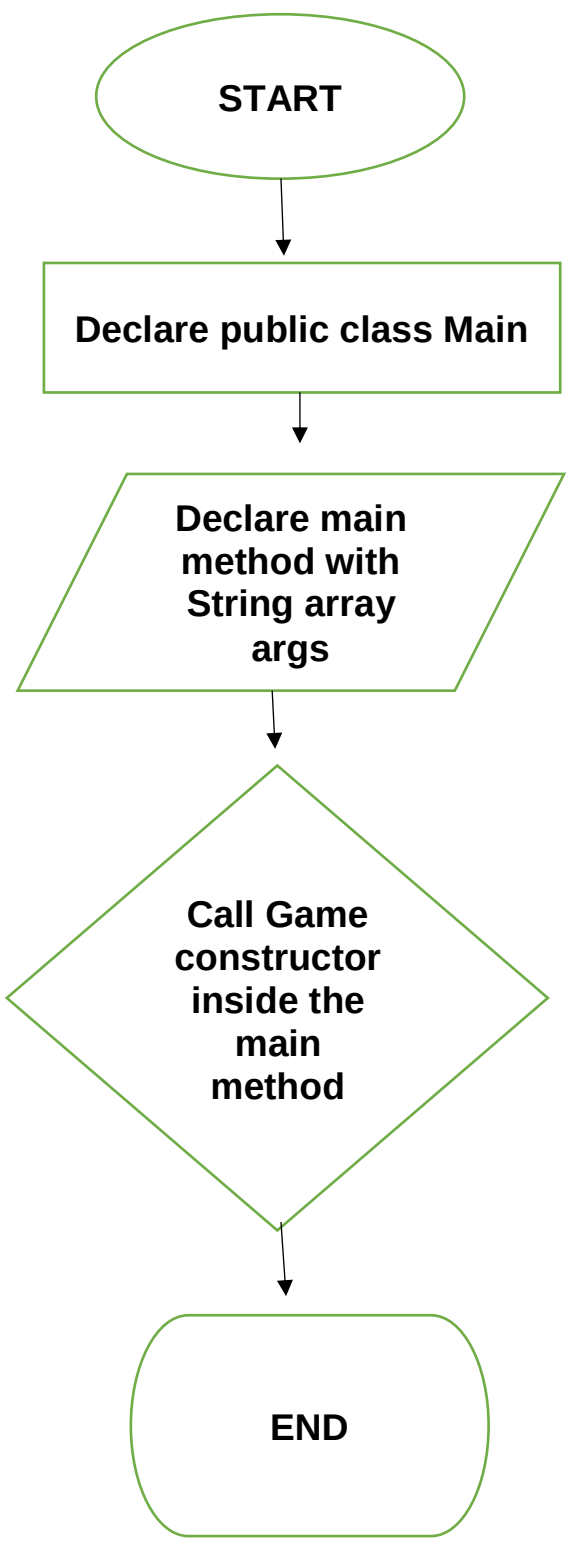
(2) FLOWCHART OF GRAPHICS CLASS:



(3) FLOWCHART OF GAME CLASS:



(4) FLOWCHART OF MAIN CLASS:



THIS IS THE ENTIRE CODE OF THE GAME

FOOD CLASS

```java
/**
 * This is our group Project
 * Members: Amninder Kaur, Khushdeep Kaur, Deepak Kumar, Ramandeep Kaur
 * Date : 31 March 2023
 * Time :10 pm
 */

import java.util.Random;  // This is the random library function

public class Food {

    private final int posX;    // these are the variables
    private final int posY;

    public Food() {    // using the function for food to set the width and height
        posX = generatePos(Graphics.WIDTH);
        posY = generatePos(Graphics.HEIGHT);
    }

    private int generatePos(int size) {  // using the random function
        Random random = new Random();
        return random.nextInt(size / Graphics.TICK_SIZE) * Graphics.TICK_SIZE;
    }

    public int getPosX() {  // using the get method
        return posX;
    }

    public int getPosY() {
        return posY;
    }
}
```

GAME CLASS

```java
/**
 * This is our group Project
 * Members: Amninder Kaur, Khushdeep Kaur, Deepak Kumar, Ramandeep Kaur
 * Date : 31 March 2023
 * Time :10 pm
 */

import javax.swing.*;  // Using the swing library

public class Game extends JFrame {  // this is a game class which is extending Jframe class

    public Game() {  // Using a game method
        this.add(new Graphics());  // using the graphics class by using add function
        this.setTitle("Snake Game");  // setting the title of game
        this.pack();  // this is a pack function
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  // to display the close button on the JFrame
        this.setResizable(true);  // setting the size of the JFrame
        this.setVisible(true);// use to show the components
        this.setLocationRelativeTo(null);
    }

}
```

GRAPHICS CLASS

```java
/**
 * This is our group Project
 * Members: Amninder Kaur, Khushdeep Kaur, Deepak Kumar, Ramandeep Kaur
 * Date : 31 March 2023
 * Time :10 pm
 */

// these are following libraries which we are using
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;

public class Graphics extends JPanel implements ActionListener {  // this is the grphics class which is
extending the JPanel class and implents the ActionListener
     // these are variables by using the final keyword
    static final int WIDTH = 700;
    static final int HEIGHT = 700;
    static final int TICK_SIZE = 50;
    static final int BOARD_SIZE = (WIDTH * HEIGHT) / (TICK_SIZE * TICK_SIZE);

    final Font font = new Font("Serif", Font.BOLD, 30);  // setting the font name,style and size
    // using the array function for the position of the snake
    int[] snakePosX = new int[BOARD_SIZE];
    int[] snakePosY = new int[BOARD_SIZE];
    int snakeLength;

    Food food;    // calling the food class
    int foodEaten;

    char direction = 'R';
    boolean isMoving = false;
    final Timer timer = new Timer(250, this);

    public Graphics() {   // method for graphics
        this.setPreferredSize(new Dimension(WIDTH, HEIGHT));  // this method is used to set the dimensions of
new screen
        this.setBackground(Color.YELLOW);  // set the background color
        this.setFocusable(true);  // using the focus function
        this.addKeyListener(new KeyAdapter() { // using the key listener function
            @Override   // the following switch case is overriding on the key listener
            public void keyPressed(KeyEvent e) {
                if (isMoving) {
                    switch (e.getKeyCode()) {  // this is our switch case statement
                        case KeyEvent.VK_LEFT:
                            if (direction != 'R') {
                                direction = 'L';
                            }
                            break;
                        case KeyEvent.VK_RIGHT:
                            if (direction != 'L') {
                                direction = 'R';
                            }
                            break;
                        case KeyEvent.VK_UP:
                            if (direction != 'D') {
                                direction = 'U';
                            }
                            break;
                        case KeyEvent.VK_DOWN:
                            if (direction != 'U') {
                                direction = 'D';
                            }
                            break;
                    }
                } else {
```

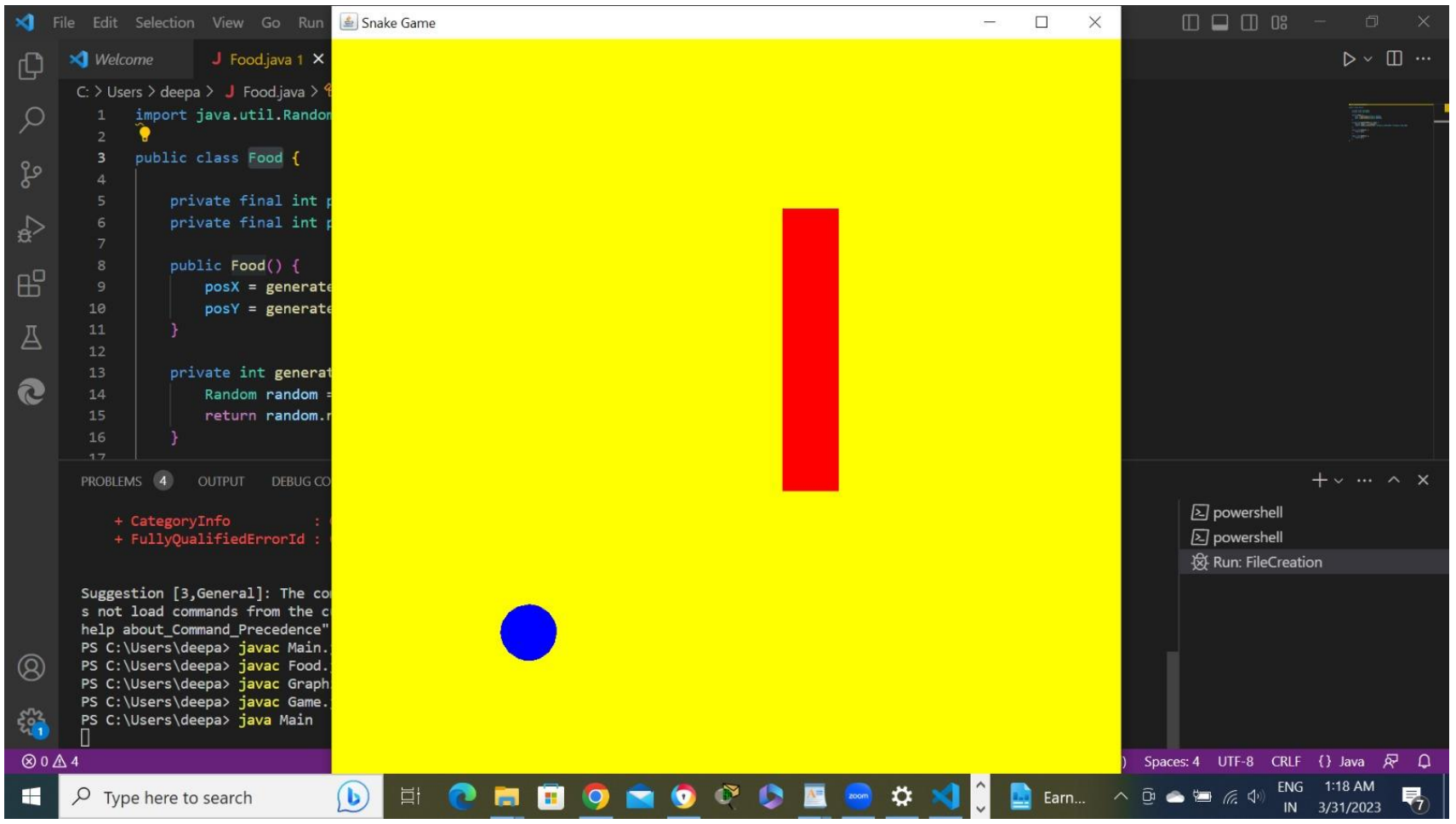
```
                start();
            }
        }
    });

    start();
}

protected void start() {   // using a start method to start the game
    snakePosX = new int[BOARD_SIZE];
    snakePosY = new int[BOARD_SIZE];
    snakeLength = 5;
    foodEaten = 0;
    direction = 'R';
    isMoving = true;
    spawnFood();
    timer.start();
}

@Override
protected void paintComponent(java.awt.Graphics g) {
    super.paintComponent(g);

    if (isMoving) {  // using the if statement for the movement of snake
        g.setColor(Color.BLUE);  // color of food
        g.fillOval(food.getPosX(), food.getPosY(), TICK_SIZE, TICK_SIZE);  // shape of food

        g.setColor(Color.RED);  // color of snake
        for (int i = 0; i < snakeLength; i++) {  // using the for loop
            g.fillRect(snakePosX[i], snakePosY[i], TICK_SIZE, TICK_SIZE);
        }
    } else {
        String scoreText = String.format("The End! Score: %d! Press any key to play again!", foodEaten);
// this is statement which would be present on the screen when th game is over

        g.setColor(Color.GREEN);
        g.setFont(font);
        g.drawString(scoreText, (WIDTH - getFontMetrics(g.getFont()).stringWidth(scoreText)) / 2, HEIGHT /
2);
    }
}

protected void move() {  // using the move method
    for (int i = snakeLength; i > 0; i--) {
        snakePosX[i] = snakePosX[i-1];
        snakePosY[i] = snakePosY[i-1];
    }

    switch (direction) {
        case 'U' -> snakePosY[0] -= TICK_SIZE;
        case 'D' -> snakePosY[0] += TICK_SIZE;
        case 'L' -> snakePosX[0] -= TICK_SIZE;
        case 'R' -> snakePosX[0] += TICK_SIZE;
    }
}

protected void spawnFood() { // method for eating the food
    food = new Food();
}

protected void eatFood() {
    if ((snakePosX[0] == food.getPosX()) && (snakePosY[0] == food.getPosY())) {
        snakeLength++;
        foodEaten++;
        spawnFood();
    }
}

protected void collisionTest() {
    for (int i = snakeLength; i > 0; i--) {
        if ((snakePosX[0] == snakePosX[i]) && (snakePosY[0] == snakePosY[i])) {
            isMoving = false;
```

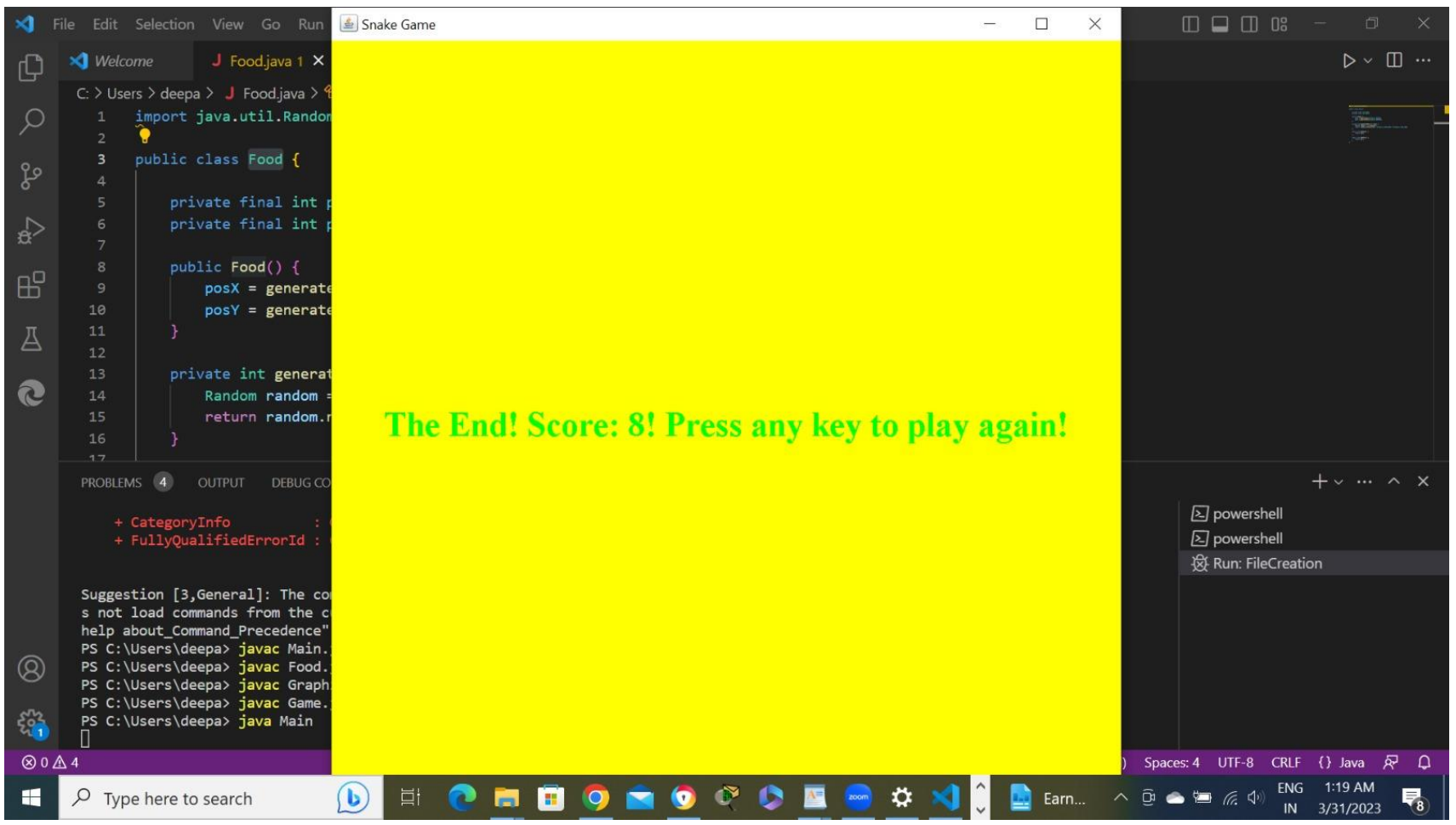
```
                break;
            }
        }

        if (snakePosX[0] < 0 || snakePosX[0] > WIDTH - TICK_SIZE || snakePosY[0] < 0 || snakePosY[0] > HEIGHT
- TICK_SIZE) {
            isMoving = false;
        }

        if (!isMoving) {
            timer.stop();
        }
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (isMoving) {
            move();
            collisionTest();
            eatFood();
        }

        repaint();
    }
}
```

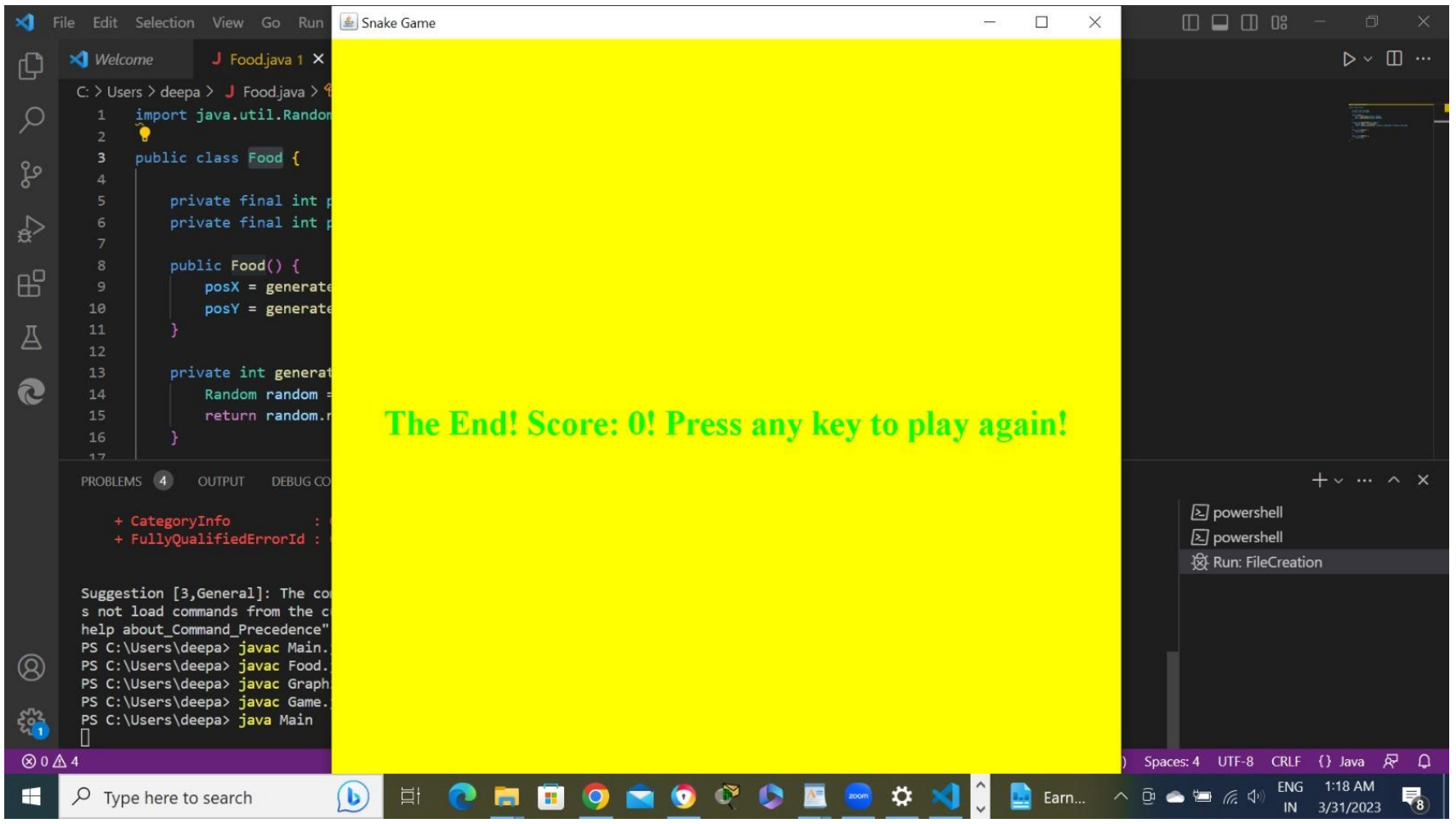
MAIN CLASS

```
/**
 * This is our group Project
 * Members: Amninder Kaur, Khushdeep Kaur, Deepak Kumar, Ramandeep Kaur
 * Date : 31 March 2023
 * Time :10 pm
 */

public class Main {   // this is our main class in which we can run the game

    public static void main(String[] args) {  // main method
        new Game();  // calling the game class
    }

}
```

OUTPUT


File Edit Selection View Go Run
Snake Game
Welcome
Food.java 1
C: > Users > deepa > Food.java >
import java.util.Rando
public class Food {
private final int
private final int
public Food() {
posX = generate
posY = generate
}
private int generat
Random random =
return random.
PROBLEMS 4 OUTPUT DEBUG CO
+ CategoryInfo :
+ FullyQualifiedErrorId :
Suggestion [3,General]: The co
s not load commands from the c
help about_Command_Precedence"
PS C:\Users\deepa> javac Main.
PS C:\Users\deepa> javac Food.
PS C:\Users\deepa> javac Graph
PS C:\Users\deepa> javac Game.
PS C:\Users\deepa> java Main
powershell
powershell
Run: FileCreation
0 4
Spaces: 4 UTF-8 CRLF {} Java
Type here to search
Earn...
ENG IN
1:18 AM 3/31/2023


File Edit Selection View Go Run
Snake Game
Welcome
Food.java 1
C: > Users > deepa > Food.java >
import java.util.Rando
public class Food {
private final int
private final int
public Food() {
posX = generate
posY = generate
}
private int generat
Random random =
return random.
PROBLEMS 4 OUTPUT DEBUG CO
+ CategoryInfo :
+ FullyQualifiedErrorId :
Suggestion [3,General]: The co
s not load commands from the c
help about_Command_Precedence"
PS C:\Users\deepa> javac Main.
PS C:\Users\deepa> javac Food.
PS C:\Users\deepa> javac Graph
PS C:\Users\deepa> javac Game.
PS C:\Users\deepa> java Main
The End! Score: 8! Press any key to play again!
powershell
powershell
Run: FileCreation
0 4
Spaces: 4 UTF-8 CRLF {} Java
Type here to search
Earn...
ENG IN
1:19 AM 3/31/2023


File Edit Selection View Go Run
Welcome
Food.java 1
C: > Users > deepa > Food.java >
import java.util.Rando
public class Food {
private final int p
private final int p
public Food() {
posX = generate
posY = generate
}
private int generat
Random random =
return random.n
}
PROBLEMS 4
OUTPUT
DEBUG CO
+ CategoryInfo :
+ FullyQualifiedErrorId :
Suggestion [3,General]: The co
s not load commands from the c
help about_Command_Precedence"
PS C:\Users\deepa> javac Main.
PS C:\Users\deepa> javac Food.
PS C:\Users\deepa> javac Graph
PS C:\Users\deepa> javac Game.
PS C:\Users\deepa> java Main
Snake Game
The End! Score: 0! Press any key to play again!
powershell
powershell
Run: FileCreation
0 4
Spaces: 4
UTF-8
CRLF
Java
Type here to search
Earn...
ENG
IN
1:18 AM
3/31/2023