MINI JAVA PROJECT

on

SNAKE GAME

Rodney Biju Reg: 12113759
Shahbaz Aman Reg: 12114313
Shahad Reg: 12114863

Submitted to
Dr. Jeevan Bala (26699)
Asst. Professor,
School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY
PHAGWARA, PUNJAB
March, 2023

Group Members and Roles:

The development of the snake game involved a team of three individuals with different roles. The team members are:

1. **Rodney Biju**: Rodney was responsible for coding the game's core logic. He developed the game's mechanics, such as snake movement, food generation, obstacle generation, and collision detection. He also implemented the game's scoring system and win/loss conditions.

2. **Shahbaz:** Shahbaz was responsible for the game's graphical elements. He created the game's graphics, such as the snake, food, and obstacles. He also implemented the snake's color-changing behaviour as the level progresses.

3. **Shahad:** Shahad was responsible for the planning phase of the project and overall code supervision. He was in charge of defining the game's objectives, requirements, and specifications. He also documented the project's progress.

System Requirements

Celeron 300 CPU expandable to PIII 1.2 Ghz.

810 E motherboard.

64 MB SD RAM, DIMM's, expandable to 256 MB.

4.3 GB HDD expandable to 80 GB and above.

10 / 100, ethernet / lan card for connecting to server, broadband --- connectivity etc.

56 Kbps fax modem voice V 90.

Sound card.

Compact keyboard.

2 stereo speakers in built.

Touchpad mouse in built in the keyboard.

3 USB 1.1 ports.

14" colour monitor, 0.28 DP, 1024 x 768 integrated.

Introduction:

The snake game is a classic arcade game that has been around for decades. It is a simple game where the player controls a snake that moves around a screen and eats food. As the snake eats food, it grows longer and faster, making it harder to avoid obstacles. In this report, we will discuss the development of a simple snake game using Java. The game starts with a snake of length 6 and speed 1, and as it eats food, the length and speed of the snake increases. The game ends if the snake hits an obstacle, and if the player reaches a snake size of 15, they win.

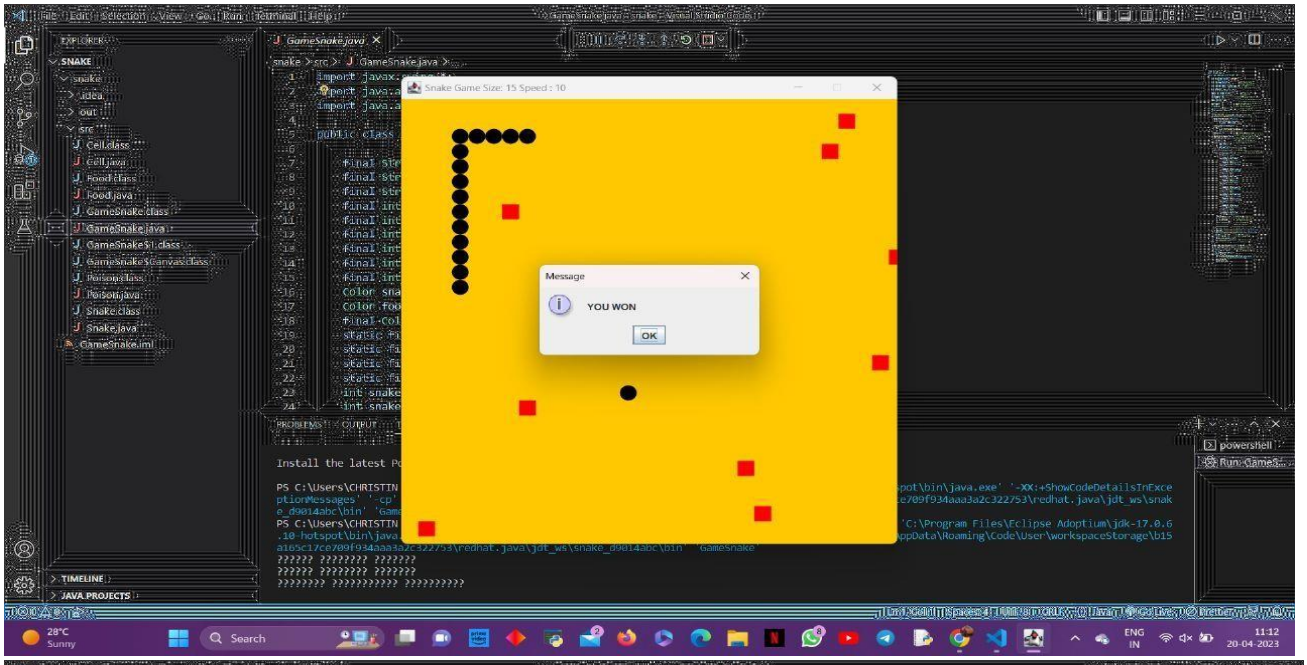
Game Development:

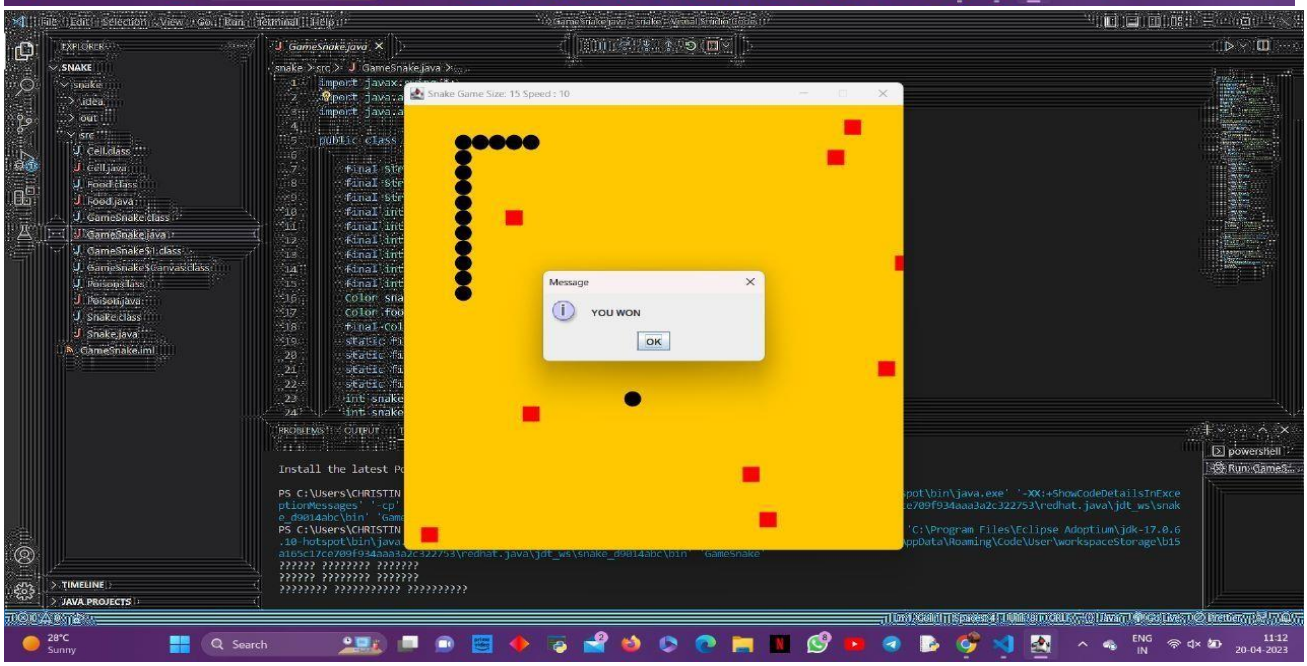
The snake game was developed using the Java programming language. The game's core logic was implemented using objectoriented programming (OOP) principles. The game's user interface was developed using Java's Swing library.

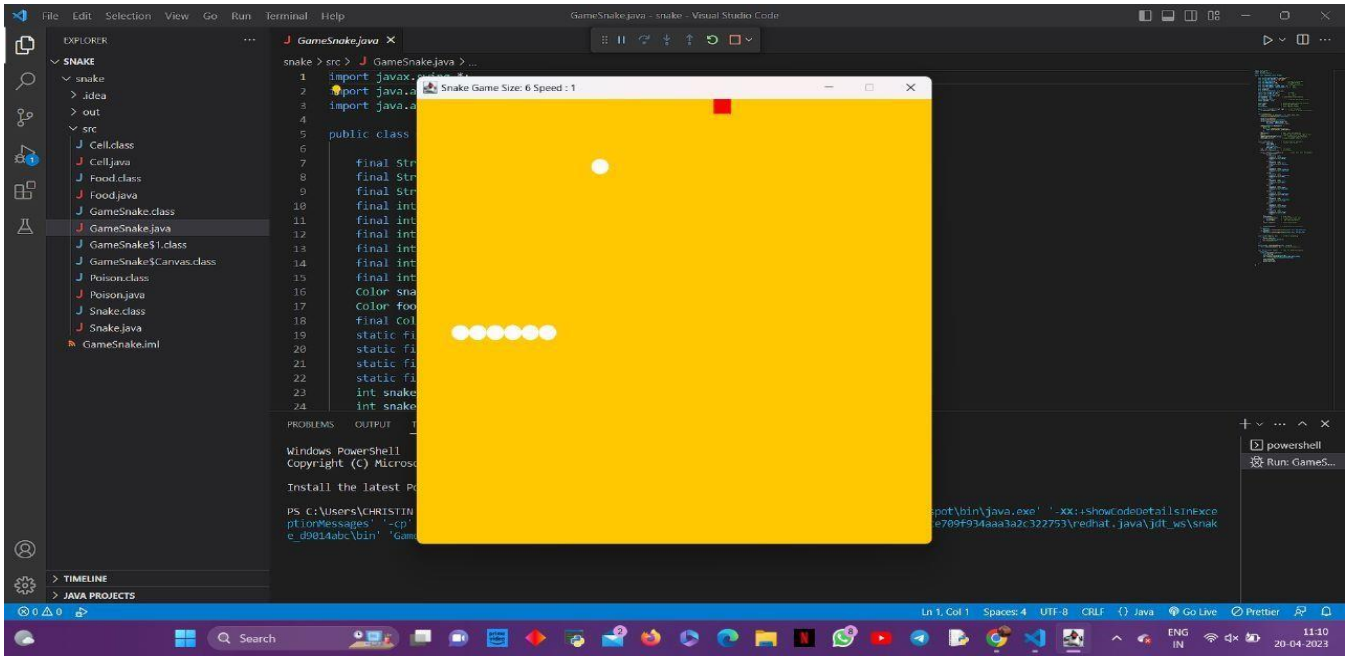
At the start of the game, the snake is displayed in white color, and the game board is generated. The snake is initialized with a length of 6 and speed of 1. Food and an obstacle are randomly generated on the game board.

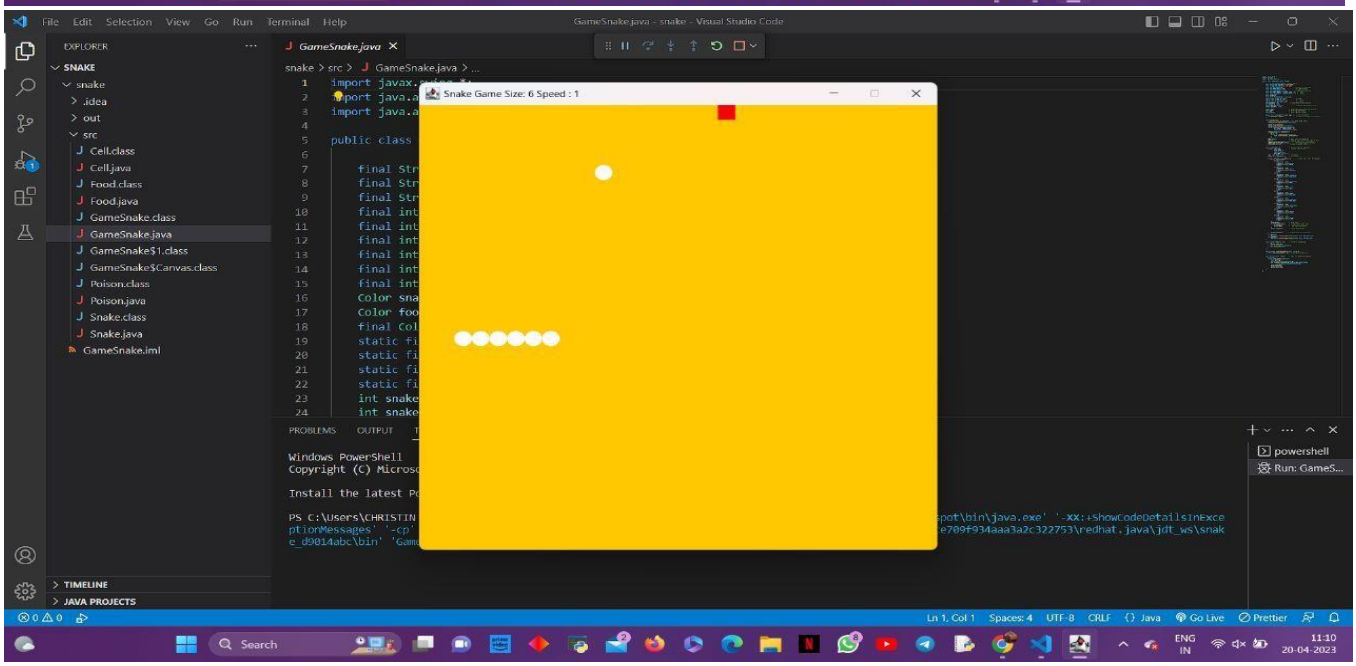
As the game progresses, the snake moves in the direction specified by the player's input. If the snake collides with food, its length increases by one, and the score is incremented. Additionally, the snake's speed increases by 0.5 units for every two foods eaten. If the snake collides with an obstacle, the game ends, and a "GAME OVER" message is displayed.

The snake's color changes as the player's score increases. The snake's color changes to grey when the score reaches 5 and to black when the score reaches 10. When the player reaches a snake size of 15, a "YOU WON" message is displayed, and the game ends.


Snake Game Size: 15 Speed : 10
Message
YOU WON
OK


Snake Game Size: 15 Speed : 10
Message
YOU WON
OK


File Edit Selection View Go Run Terminal Help
GameSnake.java - snake - Visual Studio Code
EXPLORER
SNAKE
snake
.idea
out
src
Cell.class
Cell.java
Food.class
Food.java
GameSnake.class
GameSnake.java
GameSnake$1.class
GameSnake$Canvas.class
Poison.class
Poison.java
Snake.class
Snake.java
GameSnake.iml
Snake Game Size: 6 Speed : 1
PROBLEMS
OUTPUT
Windows PowerShell
powershell
Run: GameS...
TIMELINE
JAVA PROJECTS
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Java
Go Live
Prettier
Search
ENG
IN
11:10
20-04-2023


File Edit Selection View Go Run Terminal Help
GameSnake.java - snake - Visual Studio Code
EXPLORER
SNAKE
snake
.idea
out
src
Cell.class
Cell.java
Food.class
Food.java
GameSnake.class
GameSnake.java
GameSnake$1.class
GameSnake$Canvas.class
Poison.class
Poison.java
Snake.class
Snake.java
GameSnake.iml
Snake Game Size: 6 Speed : 1
PROBLEMS
OUTPUT
Windows PowerShell
powershell
Run: GameS...
TIMELINE
JAVA PROJECTS
Ln 1, Col 1
Spaces: 4
UTF-8
CRLF
Java
Go Live
Prettier
Search
ENG
IN
11:10
20-04-2023

GameSnake.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
 public class GameSnake extends JFrame
{
     final String TITLE_OF_PROGRAM = "Snake Game";     final String
GAME_OVER_MSG = "GAME OVER";     final String GAME_WIN_MSG = "YOU
WON";     final int CELL_SIZE = 20;                  // size of cell
in pix     final int CANVAS_WIDTH = 30;               // width in
cells     final int CANVAS_HEIGHT = 30;              // height in
cells     final int START_SNAKE_SIZE = 6;            //
initialization data     final int START_SNAKE_X = CANVAS_WIDTH / 2;
//   for     final int START_SNAKE_Y = CANVAS_HEIGHT / 2; //   snake
    Color snakeColor;     Color foodColor;     final Color POISON_COLOR
= Color.red;     static final int KEY_LEFT = 37;            // codes
static final int KEY_UP = 38;              //   of     static final int
KEY_RIGHT = 39;            //   cursor     static final int KEY_DOWN =
40;             //   keys     int snakeSpeed = 1;               //
snake delay in milliseconds     int snakeDelay = 150;     boolean
gameOver = false;                   // a sign game is over or not
boolean gameWin = false;


    Canvas canvas;                // canvas object for rendering (drawing)
    Snake snake;                    // declare a snake object
    Food food;                      // declare a food object
    Poison poison;                  // declare a poison object
     public static void main(String[] args) {     // starting method
new GameSnake().game();                  // create an object and call
game()
    }
    public GameSnake() {
        //setTitle(TITLE_OF_PROGRAM + " : " + START_SNAKE_SIZE);
setDefaultCloseOperation(EXIT_ON_CLOSE);
         canvas = new
Canvas();
        canvas.setBackground(Color.ORANGE);
canvas.setPreferredSize(new Dimension(
CELL_SIZE * CANVAS_WIDTH - 10,
                CELL_SIZE * CANVAS_HEIGHT - 10));
         addKeyListener(new KeyAdapter()
{
```

```
            @Override             public void
keyPressed(KeyEvent e) {
snake.setDirection(e.getKeyCode());
            }          });          add(canvas);                    // add a
panel for rendering          pack();                        // bringing the
window to the required size           setLocationRelativeTo(null); // the
window will be in the center         setResizable(false);         // prohibit
window resizing           setVisible(true);              // make the window
visible      }       private void game() {             // method containing
game cycle          snake = new Snake(           // creating snake object
                START_SNAKE_X,
                START_SNAKE_Y,
                START_SNAKE_SIZE,
KEY_RIGHT, this);//*/          food = new Food(this);
// food object          poison = new Poison(this);
// poison object
          while (!(gameOver || gameWin)) {          // game cycle while
NOT gameOver              switch (snakeSpeed) {                 case 1:
                    snakeDelay = 150;
snakeColor = Color.WHITE;
foodColor = Color.WHITE;
break;                case 2:
                    snakeDelay = 140;
snakeColor = Color.WHITE;
foodColor = Color.lightGray;
break;                case 3:
                    snakeDelay = 130;
                    snakeColor =
Color.lightGray;                    foodColor =
Color.lightGray;                    break;
case 4:
                    snakeDelay = 125;
snakeColor = Color.lightGray;
foodColor = Color.GRAY;
break;                case 5:
                    snakeDelay = 120;
snakeColor = Color.GRAY;
foodColor = Color.GRAY;
break;                case 6:
                    snakeDelay = 115;
```

```
                    snakeColor = Color.GRAY;
foodColor = Color.DARK_GRAY;
break;                 case 7:
                    snakeDelay = 110;
snakeColor = Color.DARK_GRAY;
foodColor = Color.DARK_GRAY;
break;                 case 8:
                    snakeDelay = 105;
snakeColor = Color.DARK_GRAY;
foodColor = Color.BLACK;
break;                 case 9:
                    snakeDelay = 102;
snakeColor = Color.BLACK;
foodColor = Color.BLACK;
break;                 case 10:
                    snakeDelay = 100;
snakeColor = Color.BLACK;
foodColor = Color.BLACK;
break;
            }             snake.move();            // snake move
if (food.isEaten()) {    // if the snake ate the food
food.appear();       //   show food in new place
poison.add();        //   add new poison point
            }             canvas.repaint();        //
repaint panel/window



            sleep(snakeDelay);      // to make delay in milliseconds
        }
        if (gameOver)
            JOptionPane.showMessageDialog(GameSnake.this, GAME_OVER_MSG);
if (gameWin)
            JOptionPane.showMessageDialog(GameSnake.this, GAME_WIN_MSG);
    }      private void sleep(long ms) {    // method for
suspending         try {
            Thread.sleep(ms);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }         public boolean isCoordinatesBusy(int x, int y) {
return snake.isInSnake(x, y);// || poison.isPoison(x, y);     }
```

```
    class Canvas extends JPanel {    // class for rendering (drawing)
        @Override         public void
paint(Graphics g) {
super.paint(g);
            Graphics2D g2D = (Graphics2D) g;

g2D.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
snake.paint(g2D);             food.paint(g2D);
poison.paint(g2D);
        }
    }
}
```

Cell.java

```
import java.awt.Color; import
java.awt.Graphics2D;


public class Cell {     protected int fig;     protected int x, y;
// object coordinates     protected int size;
// object size in px     protected Color color;
// object color



    public Cell(int fig, int x, int y, int size, Color color) {
set(fig, x, y);                              // init coordinates
this.size = size;                       // init size         this.color
= color;                   // init color     }
```

```
    public void set(int fig, int x, int y) {           // set
coordinates         this.fig = fig;         this.x = x;         this.y =
y;
    }      public int getX() {                      // get the X
coordinate         return x;
    }      public int getY() {                      // get the Y
coordinate         return y;
    }      public void paint(Graphics2D g) {          // object
rendering         g.setColor(color);         switch (fig) {
case 2:
               g.fillRect(x * size, y * size,        // upper left
corner                          size, size);                break;//
width and height            case 1:
               g.fillOval(x * size, y * size,        // upper left
corner                          size, size);                break;//
width and height
        }
    }
}
```

Food.java

```
import java.util.Random; public
class Food extends Cell {
private GameSnake gameSnake;
private Random random;


    public Food(GameSnake gameSnake) {      // constructor
super(1, -1, -1, gameSnake.CELL_SIZE, gameSnake.foodColor);
this.gameSnake = gameSnake;         random = new Random();
    }       public boolean isFood(int x, int y) {    // if food has x, y
coordinate         return (this.x == x) && (this.y == y);
    }      public boolean isEaten() {              // if food is eaten
or not
```

```
        return getX() == -1;
    }      public void eat() {                    // virtual
eating          set(1, -1, -1);
    }      public void appear() {                 // show food at
new x,y coordinates          int x, y;          do {             x =
random.nextInt(gameSnake.CANVAS_WIDTH);             y =
random.nextInt(gameSnake.CANVAS_HEIGHT);         } while
(gameSnake.isCoordinatesBusy(x, y));          set(1, x, y);
    }
}
```

Conclusion:

The development of the snake game involved a team of three individuals with different roles. The game's planning phase was

handled by Shahad, while Rodney was responsible for coding the game's core logic. Shahbaz was responsible for the game's graphical elements. The snake game was developed using Java and involved the use of OOP principles and Java's Swing library for the game's user interface. The game's mechanics include snake movement, food generation, obstacle generation, collision detection, scoring system, and win/loss conditions. The snake's color changes as the player's score increases, and the game ends if the snake hits an obstacle or reaches a size of 15.