

프로그래밍 언어응용

문제해결 시나리오

AI기반 스마트제조공정 빅데이터 분석 시스템 개발 1차

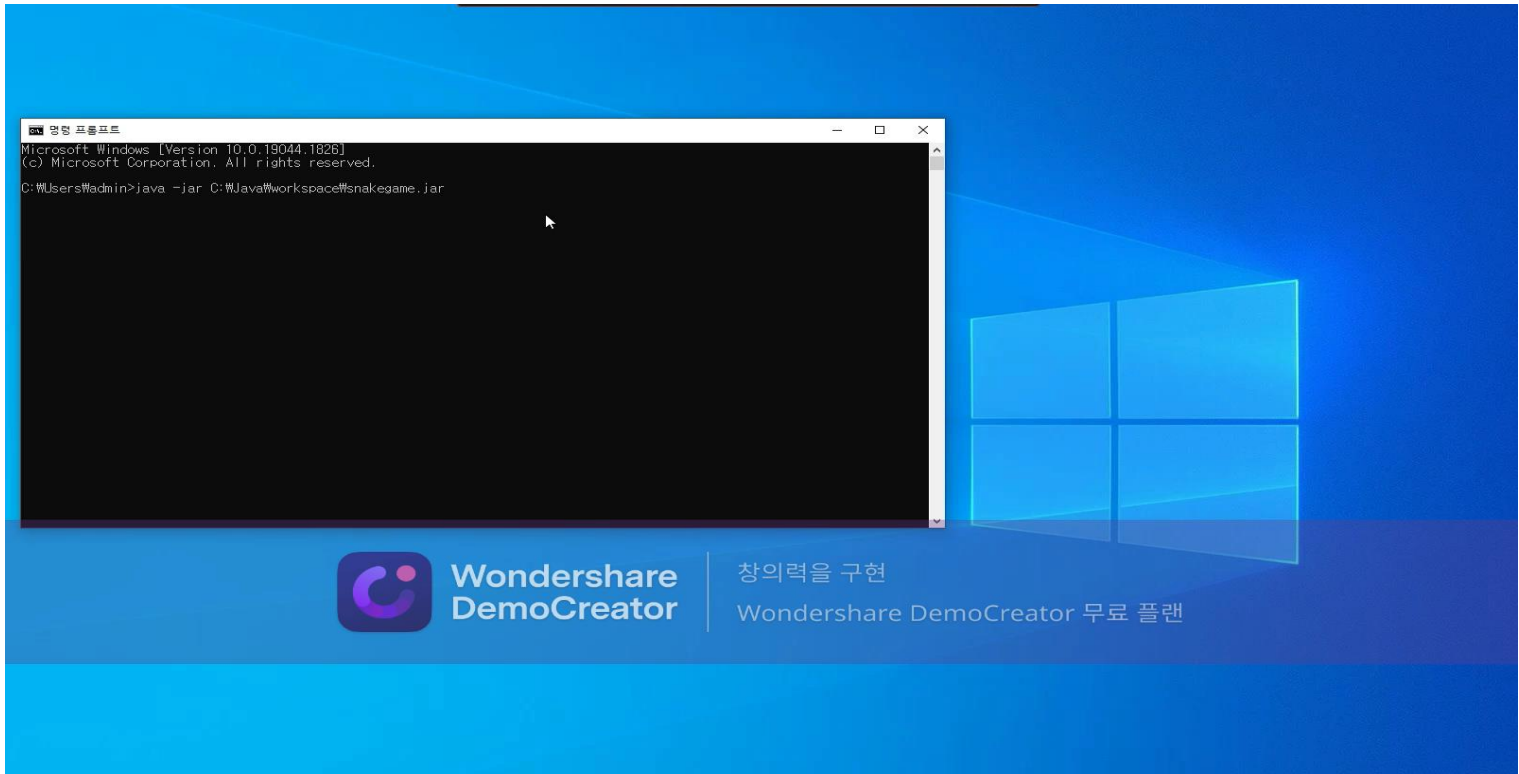
장 준 혁

I N D E X

1. Snake Game 결과 영상	3
2. 클래스 구성하기.....	4
3. 메인 클래스 코드 작성하기	5
4. 프레임 클래스 코드 작성하기	7
5. 패널 클래스 코드 작성하기	9
6. 참고 자료	17

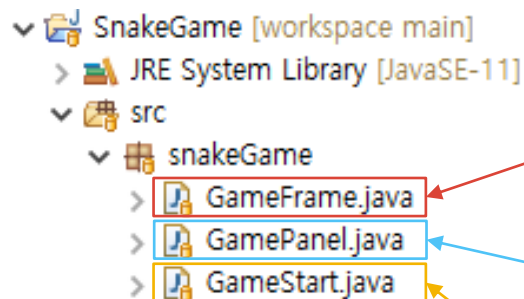


1. Snake Game 영상



2. 클래스 구성하기

- 클래스는 3개로 구성하였습니다.



Swing의 프레임창을
지정해주는 클래스

플레이어의 조작에 게임이
동작하도록 설정한 클래스

게임의 난이도를 선택하고
시작하는 **메인** 클래스

3. 메인 클래스 코드 작성하기 (1)

- 게임의 시작화면을 출력합니다.

```
import java.util.Scanner;

public class GameStart {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("=== 게임을 시작합니다. 난이도를 선택해주세요. ===");
        System.out.println("    [1]EASY    [2]NORMAL    [3]HARD    ");
        System.out.println("=====");
```

3. 메인 클래스 코드 작성하기 (2)

- 게임의 난이도를 선택합니다.

```
/* 난이도 선택하기 : 1~3번 키입력이 아닐 경우 해당 키가 입력될 때까지 반복한다. */
boolean isNumber = false;

do {
    String level = scan.nextLine();

    try {
        int number = Integer.parseInt(level);

        switch (level) {
            case "1":
                System.out.println("Easy Mode를 선택하셨습니다.");
                new GameFrame(1);
                break;
            case "2":
                System.out.println("Normal Mode를 선택하셨습니다.");
                new GameFrame(2);
                break;
            case "3":
                System.out.println("Hard Mode를 선택하셨습니다.");
                new GameFrame(3);
                break;
            default:
                System.out.println(number + "번은 올바른지 않은 선택이니 1~3번 중에 다시 입력해 주세요.");
        }
    } catch (NumberFormatException e) {
        System.out.println("숫자를 입력해주세요.");
    }
} while (!isNumber);

scan.close();
}
```

do while 반복문과
try catch 문을 이용하여
1, 2, 3 정수 값만 받도록
코드를 작성

4. 프레임 클래스 코드 작성하기 (1)

- 플레이어가 선택한 난이도에 따라 게임을 실행합니다.

```
import javax.swing.JFrame;

@SuppressWarnings("all") // 모든 경고창 무시하기
public class GameFrame extends JFrame {

    private static final long serialVersionUID = 1L;
    private int mode = 1;

    /* 플레이어가 선택한 난이도에 따라 게임이 실행됨 */
    public GameFrame(int mode) {

        this.mode = mode;

        if (mode == 1) {
            add(new JPanel(80, 5));
        }
        if (mode == 2) {
            add(new JPanel(60, 6));
        }
        if (mode == 3) {
            add(new JPanel(40, 7));
        }
    }
}
```

EASY

NORMAL

HARD

클래스 내에서
변수를 사용하지 않았다고
경고가 발생하나
실행에는 문제가 없음

4. 프레임 클래스 코드 작성하기 (2)

- 게임 화면의 프레임창을 설정합니다.

```
/* 프레임창 설정 */
setTitle("Enjoy~ Snake Game!");
setVisible(true);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setResizable(false);
pack(); // 구성 요소들의 크기를 화면 크기에 맞춰 자동으로 조절
setLocationRelativeTo(null); // 컴퓨터 화면 중앙에 창을 띄움
```

```
}
```

```
}
```


5. 패널 클래스 코드 작성하기 (1)

- 게임에 사용될 변수를 설정합니다.

```
import java.awt.*;
import java.awt.event.*;
import java.util.Random;
import javax.swing.*;

public class GamePanel extends JPanel implements ActionListener {

    private static final long serialVersionUID = 1L;

    /* 변수 설정하기 */
    private int screenWidth = 1500;
    private int screenHeight = 900;
    private int gridSize = 60;
    private int grids = (screenWidth / gridSize) * (screenHeight / gridSize);
    private int x[] = new int[grids];
    private int y[] = new int[grids];

    private int foodsEaten = 0;
    private int foodX;
    private int foodY;

    private char direction = 'R';
    private boolean running = false;

    public int speed;
    public int bodyLength;

    Timer timer;
    Random random = new Random();
```

생성자로 사용(난이도와 관련)되어
다른 클래스에서도 이용하므로
public으로 설정

5. 패널 클래스 코드 작성하기 (2)

- 생성자 및 게임 시작과 관련된 메소드들을 설정합니다.

/* 패널화면 설정 */

```
GamePanel(int speed, int bodyLength) {
```

```
    this.speed = speed;
```

```
    this.bodyLength = bodyLength;
```

```
    setBackground(Color.BLACK);
```

```
    setPreferredSize(new Dimension(screenWidth, screenHeight));
```

```
    setFocusable(true);
```

```
    addKeyListener(new MyKeyAdapter());
```

```
    startGame();
```

```
}
```

```
public void startGame() {
```

```
    newFood();
```

```
    running = true;
```

```
    timer = new Timer(speed, this);
```

```
    timer.start();
```

```
}
```

/* 새로운 음식이 화면에 랜덤으로 나타나게 하기(보이지는 않음) */

```
public void newFood() {
```

```
    foodX = random.nextInt(screenWidth / gridSize) * gridSize;
```

```
    foodY = random.nextInt(screenHeight / gridSize) * gridSize;
```

```
}
```

```
@Override
```

```
public void paintComponent(Graphics g) {
```

```
    super.paintComponent(g); // 패널에서 이전에 그려진 잔상을 지워줌
```

```
    draw(g);
```

```
}
```

5. 패널 클래스 코드 작성하기 (3)

- 음식과 뱀(플레이어)을 화면에 표시해주는 메소드를 설정합니다.

```
public void draw(Graphics g) {  
    if (running) {  
        /* 화면에 격자 표시하기 */  
        for (int i = 0; i < screenWidth / gridSize; i++) {  
            g.drawLine(0, i * gridSize, screenWidth, i * gridSize); // 가로 격자  
            g.drawLine(i * gridSize, 0, i * gridSize, screenHeight); // 세로 격자  
        }  
  
        /* 음식을 화면에 보이게 하기 */  
        g.setColor(new Color(random.nextInt(256), random.nextInt(256), random.nextInt(256)));  
        g.fillRect(foodX, foodY, gridSize, gridSize);  
  
        /* 뱀(플레이어)을 화면에 나타내기 */  
        for (int i = 0; i < bodyLength; i++) {  
            if (i == 0) {  
                g.setColor(Color.GREEN);  
                g.fillRect(x[i], y[i], gridSize, gridSize);  
            } else {  
                g.setColor(Color.WHITE);  
                g.fillRect(x[i], y[i], gridSize, gridSize);  
            }  
        }  
  
        /* 화면 가운데 상단에 점수 표시하기 */  
        g.setColor(Color.WHITE);  
        g.setFont(new Font("SanSerif", Font.BOLD, 30));  
        FontMetrics metrics1 = getFontMetrics(g.getFont());  
        g.drawString("Score : " + foodsEaten, (screenWidth - metrics1.stringWidth("Score : " + foodsEaten)) / 2,  
                    gridSize);  
    } else {  
        gameOver(g);  
    }  
}
```

gridSize에 맞게 음식과 뱀을
화면에 사각형으로 채워준다.

5. 패널 클래스 코드 작성하기 (4)

- 게임 종료와 관련된 메소드를 설정합니다.

```
public void gameOver(Graphics g) {  
    /* 점수 표시하기 */  
    g.setColor(Color.WHITE);  
    g.setFont(new Font("SanSerif", Font.BOLD, 30));  
    FontMetrics metrics2 = getFontMetrics(g.getFont());  
    g.drawString("Score : " + foodsEaten, (screenWidth - metrics2.stringWidth("Score : " + foodsEaten)) / 2,  
        gridSize);  
  
    /* 게임종료 표시하기 */  
    g.setColor(Color.RED);  
    g.setFont(new Font("SanSerif", Font.BOLD, 80));  
    FontMetrics metrics3 = getFontMetrics(g.getFont());  
    g.drawString("Game Over", (screenWidth - metrics3.stringWidth("Game Over")) / 2, screenHeight / 2);  
}
```

5. 패널 클래스 코드 작성하기 (5)

- 뱀이 움직이는 메소드를 설정합니다.

```
public void move() {
    for (int i = bodyLength; i > 0; i--) {
        x[i] = x[i - 1];
        y[i] = y[i - 1];
    }

    /* U → Up, D → Down, L → Left, R → Right */
    switch (direction) {
        case 'U':
            y[0] = y[0] - gridSize;
            break;
        case 'D':
            y[0] = y[0] + gridSize;
            break;
        case 'L':
            x[0] = x[0] - gridSize;
            break;
        case 'R':
            x[0] = x[0] + gridSize;
            break;
    }
}
```

5. 패널 클래스 코드 작성하기 (6)

- 키 입력을 받아서 뱀을 조정할 수 있게 해줍니다.

```
public class MyKeyAdapter extends KeyAdapter {
    @Override
    public void keyPressed(KeyEvent e) {
        int keyCode = e.getKeyCode(); // 상하좌우 방향키는 유니코드가 아니므로 getKeyCode() 사용
        switch (keyCode) {
            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;
        }
    }
}
```

5. 패널 클래스 코드 작성하기 (7)

- 뱀이 음식을 먹거나 충돌했을 때의 메소드를 설정합니다.

```
/* 뱀이 음식을 먹었을 때 */
public void eatFood() {
    if ((x[0] == foodX) && (y[0] == foodY)) {
        bodyLength++;
        foodsEaten++;
        newFood();
    }
}

/* 뱀이 충돌했을 때 */
public void collisions() {
    /* 머리가 자신의 몸이랑 충돌했을 때 */
    for (int i = bodyLength; i > 0; i--) {
        if (x[0] == x[i] && y[0] == y[i]) {
            running = false;
        }
    }

    /* 벽에 충돌했을 때 */
    if ((x[0] < 0) || (x[0] > screenWidth) || (y[0] < 0) || (y[0] > screenHeight)) {
        running = false;
    }

    if (!running) {
        timer.stop();
    }
}
```

5. 패널 클래스 코드 작성하기 (8)

- 게임이 진행 중일 때 위에서 설정한 메소드들이 계속 수행되게 합니다.

```
@Override
public void actionPerformed(ActionEvent e) {
    if (running) {
        move();
        eatFood();
        collisions();
    }
    repaint();
}
}
```


참 고 자 료

1. <https://www.youtube.com/watch?v=bl6e6qjJ8JQ>
2. <https://www.youtube.com/watch?v=7t5uwECup4I>
3. <https://zetcode.com/javagames/snake/>



감사합니다.