



2021-1 객체지향프로그래밍 30조

IT공학전공 1914386 김민지


IT공학전공 2016680 김민지

IT공학전공 1914248 노현진

IT공학전공 2012918 오예지



농송이는



배고프숙



CONTENTS



주제 및
선정 배경



전반적인
프로젝트 구성



세부적인
기능 소개



맡은 역할
및 기여도



실시간
시연



LEVEL 1

주제 및 선정 배경

눈송이는 배고프속



뷰어스 | 2020.12.23.
게임계에도 '뉴트로' 열풍...왜 고전게임에 열광하나
뷰어스 송인화 기자 신세대를 쫓는 '뉴트로' 열풍이 게임계에도 불고 있다. 뉴트로
는 새로움(New)과 복고(Retro)를 합친 신조어로 복고를 새롭게 즐기는 경향을 말...

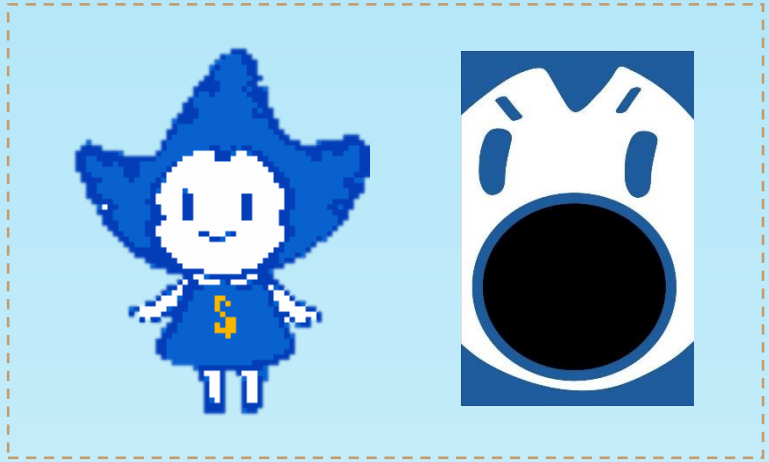
"코로나19 이후 1년, 게임 '심리방역' 역할 수행"

이주환 기자 | 2021.04.30 11:09 | 댓글 0

매일경제 | 2020.09.19 | 네이버뉴스
뉴트로 열풍에 도트 그래픽 모바일게임 '눈길'
문화·예술 전반에서 보여지는 뉴트로 열풍이 게임 시장에서도 위력을 발휘하고 있
다. 도트 그래픽은 사각형의 작은 점으로 화면을 표현한 것이 특징이다. 과거 아...

산업 > 바이오&ICT
싸이월드·버디버디·포트리스·디아블로... IT도 '뉴트로 열풍'

입력 2021-03-07 17:42:19 수정 2021.03.08 14:06:31 문민혁 기자



<눈송이 캐릭터가 하늘에서 떨어지는 음식을 받아먹는 게임>

최근 뉴트로 열풍이 불며 도트 그래픽을 활용한 고전 게임이 다시 주목을 받고 있음.

추억의 게임을 재현함으로써 어릴 적으로 돌아가 추억회상을 할 수 있도록 본 게임을 기획함.

또한, 코로나 19 이후 게임의 수요가 급증함에 따라 집에서 즐길 수 있는 게임을 제작하고자 함.

전반적인 프로젝트 구성

눈송이 캐릭터가 하늘에서 떨어지는 음식을 받아먹는 게임



눈송이

- ← → 방향키 사용
- 하늘에서 떨어지는 음식을 먹음
- 음식을 먹으면 점수 증가
- 폭탄을 먹으면 점수 감소 & 하트 감소
- 생명 소진 시 게임 종료 & 최종 순위

H



음식

- 음식 종류 & 생성 위치 & 개수 랜덤
- 음식을 먹으면 점수 증가

F

+10 +20 +30



폭탄

- 생성 위치 & 개수 랜덤
- 폭탄을 먹으면 점수 & 생명 감소

B

-50

&

-



사용한 클래스 & 클래스들 간의 관계

눈송이는 배고프속

<사용한 클래스>

- Java.awt : 그래픽, 이미지 및 폰트 출력 등 구현에 전반적으로 사용되는 클래스
- Java.awt.event : 키보드 이벤트, 버튼 이벤트를 위해 사용한 클래스
- javax.swing : JFrame, JPanel 등 구현에 전반적으로 사용한 클래스
- Java.io : 입출력을 위한 클래스
- Java.util : 랜덤 함수, ArrayList 사용을 위한 클래스
- javax.sound.sampled : 배경음악을 위해 사용한 클래스
- Java.sql : sql 사용을 위한 클래스

배운 클래스

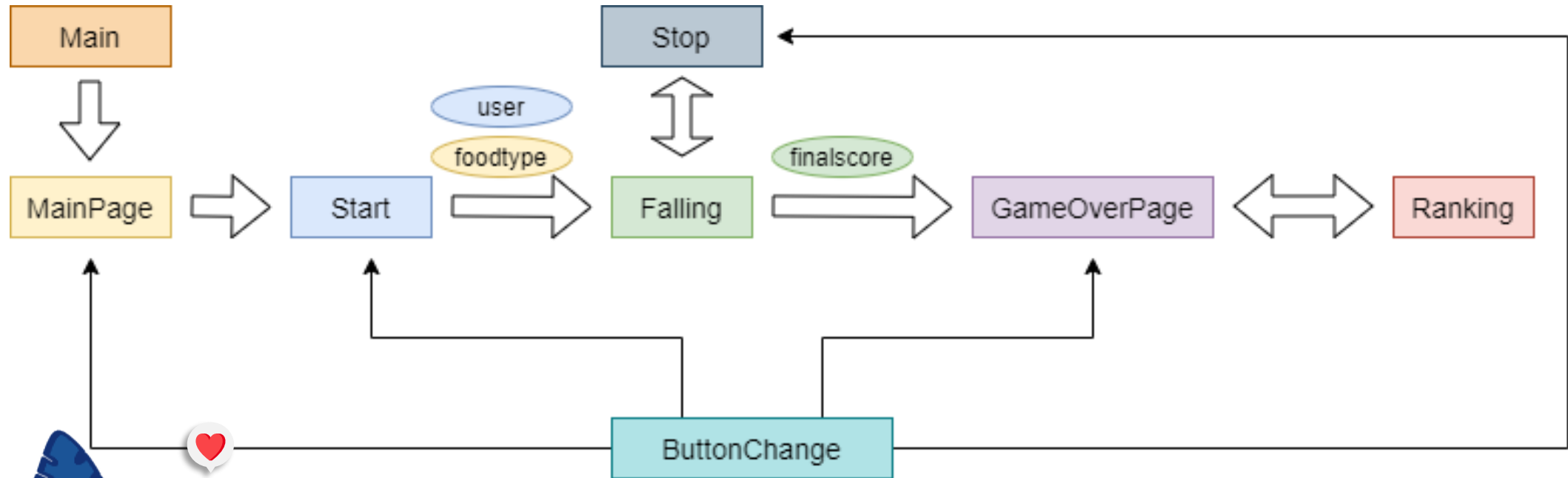
새로운 클래스



LEVEL
2

사용한 클래스 & 클래스들 간의 관계

눈송이는 배고프속



LEVEL

3

세부적인 기능 소개

Main.java

게임 전체에 "둥근모퉁" 폰트를 사용하고 있어서 게임을 실행하기 전, 폰트 파일을 다운로드해 주세요!

```
// 폰트 -----  
File font_file = new File("neodgm.ttf");  
Font font = Font.createFont(Font.TRUETYPE_FONT, font_file);  
Font font_s = font.deriveFont(20f); // 작은 폰트  
Font font_l = font.deriveFont(40f); // 큰 폰트
```

1. 데이터베이스 연결, MainPage 클래스 실행

```
public static void main (String args[]) throws FontFormatException, IOException {  
    statement = checkdb();  
    MainPage mp = new MainPage();  
    mp.setVisible(true);  
}  
  
public static Statement checkdb() {  
    Connection con = null;  
    Statement st = null;  
  
    try{  
        Class.forName("com.mysql.cj.jdbc.Driver");  
        String url = "jdbc:mysql://us-cdb-east-04.cleardb.com/heroku_f91ffea5f7054c5";  
        con = DriverManager.getConnection(url, "bcb459354f7338", "eaaea278");  
        st = con.createStatement();  
        System.out.println("연결 성공");  
    }  
    catch(ClassNotFoundException e){  
        System.out.println("드라이버 로딩 실패");  
    }  
    catch(SQLException e){  
        System.out.println("에러: " + e);  
    }  
    return st;  
}
```

- Main class에서는 메인함수로 MainPage를 실행시킴

- 추후 랭킹을 보여주기 위해 데이터베이스에 연결하는데 Main class에서 데이터베이스 연결을 체크하여 연결 성공 여부를 출력함.

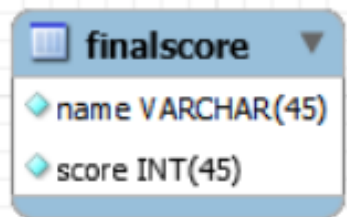
LEVEL

3

세부적인 기능 소개

데이터베이스 (데이터베이스 연동 버전)

1. 입력받은 사용자의 이름과 게임 점수를 점수를 출력하기 위해 데이터베이스를 사용
2. heroku라는 외부 서버에서 데이터베이스를 만들어, 모든 사용자가 따로 DB를 구축하지 않아도 되도록 함
3. 문자열 타입의 name과 정수형 타입의 score을 테이블의 열로 설정
4. Main.java 의 **checkdb()** 메소드를 통해 데이터베이스를 연동



```
try{
    Class.forName("com.mysql.cj.jdbc.Driver");
    String url = "jdbc:mysql://us-cdbr-east-04.cleardb.com/heroku_f91ffea5f7054c5";
    con = DriverManager.getConnection(url, "bcb459354f7338", "eaaea278");
    st = con.createStatement();
    System.out.println("연결 성공");
}
catch(ClassNotFoundException e){
    System.out.println("드라이버 로딩 실패");
}
catch(SQLException e){
    System.out.println("에러: " + e);
}
return st;
```


세부적인 기능 소개

스코어 파일 - FileHelp.java / Ranking.java (파일 입출력 버전)

1. 입력받은 사용자의 이름과 게임 점수를 점수를 출력하기 위해 파일입출력 사용

```
public Ranking() {  
    file = new File("userscores.txt");  
}
```

2. userscores.txt 파일을 이용해 유저 이름과 점수를 기록

3. 두 객체씩 비교하며 음수인 경우 객체의 자리를 유지하고

양수인 경우 두 객체의 자리를 바꾸면서 내림차순 정렬

```
public void save() throws IOException { // 저장  
    FileWriter fw = new FileWriter(file);  
    BufferedWriter bw = new BufferedWriter(fw);  
    PrintWriter pw = new PrintWriter(bw);  
  
    for (FileHelp save : sclist) {  
        pw.println(save.getName() + "@" + save.getScore());  
        pw.flush();  
    }  
    pw.close();  
}
```

```
protected void rank() {  
    FileHelp[] rank = new FileHelp[sclist.size()];  
    sclist.toArray(rank);  
    Collections.reverse(sclist);  
  
    for (int i=0; i<rank.length;i++) {  
        for (int j=0; j<rank.length; j++) {  
            if (rank[i].getScore() < rank[j].getScore()) {  
                rank[i].plusRank();  
            }  
        }  
    }  
}  
  
public void scoreprint() { //순위별로 출력  
    sclist.sort(new Comparator<FileHelp>() {  
  
        @Override  
        public int compare(FileHelp o1, FileHelp o2) {  
            // TODO Auto-generated method stub  
            int score1 = o1.getScore();  
            int score2 = o2.getScore();  
            if (score1 == score2) {  
                return 0;  
            } else if (score1 > score2) {  
                return 1;  
            } else {  
                return -1;  
            }  
        }  
    });  
    rank();  
  
    for (FileHelp output : sclist) {  
        nameList.add(output.getName());  
        scoreList.add(output.getScore());  
        output.totalprint();  
    }  
}
```

→ 정렬 코드

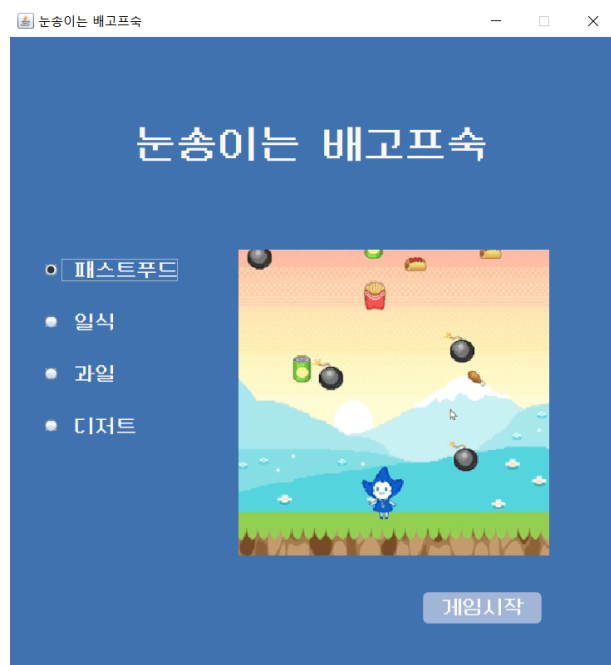
LEVEL

3

세부적인 기능 소개

메인 화면 - MainPage.java

1. 시작하기 전에 **음식 종류** 선택
클릭 시, 해당하는 음식으로 게임을 하는 프리뷰 이미지 보임
JRadioButton의 옵션에 따라 떨어질 음식 종류(foodtype)가 결정됨



```
option = new JRadioButton[4];
ButtonGroup options = new ButtonGroup();

option[0] = new JRadioButton(" 패스트푸드", true);
option[1] = new JRadioButton(" 일식");
option[2] = new JRadioButton(" 과일");
option[3] = new JRadioButton(" 디저트");
```

```
// 옵션 별로 프리뷰 이미지 다르게 설정
@Override
public void actionPerformed(ActionEvent e) {
    if (option[0].isSelected()==true) {
        imglabel.setIcon(new ImageIcon("Images/fastfoodpre.gif"));
        foodtype = "fastfood";
    } else if (option[1].isSelected()==true) {
        imglabel.setIcon(new ImageIcon("Images/japanesepre.gif"));
        foodtype = "sushi";
    } else if (e.getSource() == option[2]) {
        imglabel.setIcon(new ImageIcon("Images/fruitpre.gif"));
        foodtype = "fruit";
    } else if (e.getSource() == option[3]) {
        imglabel.setIcon(new ImageIcon("Images/dessertpre.gif"));
        foodtype = "dessert";
    }

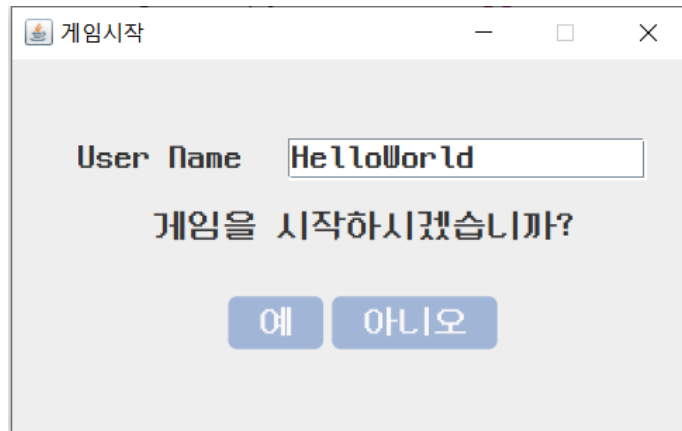
    if (e.getSource() == startbt) {
        s.setVisible(true);
    }
}
```

세부적인 기능 소개

시작 화면 - Start.java

actionPerformed: 버튼 이벤트 리스너

- MainPage에서 선택한 음식 종류(foodtype)를 게임(Falling) 클래스로 전달
- '예' 버튼을 누르면 게임(Falling) 클래스가 열림
- '아니오' 버튼을 누르면 창이 닫힘



```
public void actionPerformed(ActionEvent e) {
    user = name.getText(); // textfield에서 입력받은 값 가져오기
    if(e.getSource() == yes) {
        try {
            MainPage mp = new MainPage(); // foodtype 받아오기 위해
            Falling f2 = new Falling(mp.foodtype); // 게임에 foodtype 넘겨줌
            f2.setVisible(true);
        } catch (FontFormatException | IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
        setVisible(false);
    } else if(e.getSource() == no) {
        dispose();
        setVisible(false);
    }
}
```

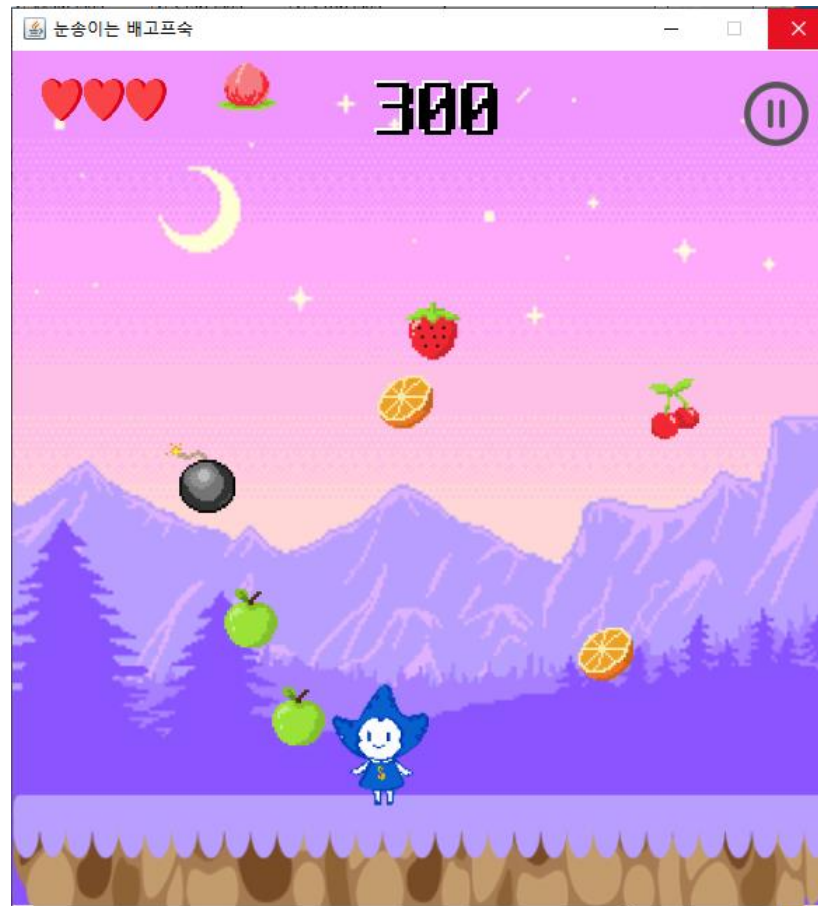
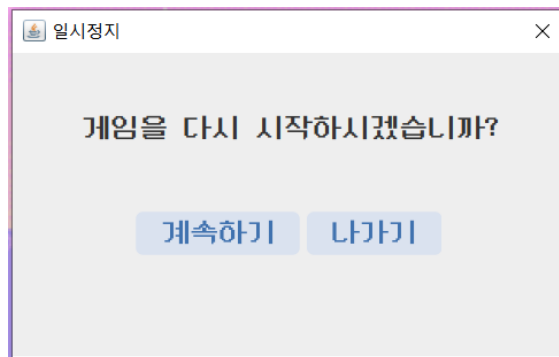
LEVEL

3

세부적인 기능 소개

본 게임 - Falling.java

1. 키보드의 좌,우 방향키를 받아 눈송이를 움직임
2. 화면 상단에 점수, 현재의 하트 수, 일시정지 버튼
3. 음식 먹으면 (10, 20, 30)점 중 랜덤으로 점수 상승
폭탄 먹으면 점수 50점 감소, 하트 한 개 감소
4. 일시정지 창 계속하기 버튼 -> 게임 이어서 시작,
나가기 버튼 -> 메인화면으로 돌아감



세부적인 기능 소개

본 게임 - Falling.java

<주요 함수>

1. Falling: JFrame을 상속하는 생성자

- JPanel, GameThread 실행
- 눈송이 움직이는 keyListener

2. loadAudio: 배경음악 생성

```
panel = new JPanel();  
add(panel, BorderLayout.CENTER);
```

```
gThread = new GameThread(); // Thread 생성  
gThread.start(); // Thread 시작하여 run() 메소드 자동 실행
```

```
loadAudio("retrogame.wav"); // 오디오  
clip.start();
```

```
addKeyListener(new KeyListener() {  
    @Override  
    public void keyPressed(KeyEvent e) {  
        int keyCode = e.getKeyCode();  
        switch (keyCode) {  
            case KeyEvent.VK_LEFT: // 왼쪽 키를 누르면 눈송이 x좌표가 -10씩 이동함.  
                panel.x -= 10;  
                break;  
            case KeyEvent.VK_RIGHT:  
                panel.x += 10; // 오른쪽 키를 누르면 눈송이 x좌표가 +10씩 이동함.  
                break;  
        }  
    }  
});
```

```
private void loadAudio(String pathName) { // 오디오 설정  
    try {  
        clip = AudioSystem.getClip();  
        File audioFile = new File(pathName);  
        AudioInputStream audioStream = AudioSystem.getAudioInputStream(audioFile);  
        clip.open(audioStream);  
        clip.loop(clip.LOOP_CONTINUOUSLY); // 반복 재생  
    }  
    catch (LineUnavailableException e) {e.printStackTrace();}  
    catch (UnsupportedAudioFileException e) {e.printStackTrace();}  
    catch (IOException e) {e.printStackTrace();}  
}
```



세부적인 기능 소개

본 게임 - Falling.java

<주요 함수>

3. **GamePanel**: JPanel 상속, 실질적인 게임 화면 출력

- 음식, 폭탄, 생명 이미지는 유동적이어야 하므로 ArrayList에 저장
- paintComponent: 배경, 눈송이, 음식, 폭탄, 생명 이미지 그리기
 - MainPage에서 전달받은 foodtype은 파일 이름으로 들어감

```
for (int i = 0; i < 7; i++) { // 음식 이미지
    foodarr.add(Toolkit.getDefaultToolkit().getImage("Images/" + Foodtype + i + ".png"));
}
```

- move(): 눈송이 이동

```
void move() {
    x += dx;
    // 눈송이 이미지가 화면 밖으로 나가지 않도록 설정
    if (x < w) // 왼쪽 끝
        x = w;
    if (x > width - w) // 오른쪽 끝
        x = width - w;
}
```

〈주요 함수〉

3. **GamePanel** : JPanel 상속, 실질적인 게임 화면 출력

- makeFood, makeBomb: 음식, 폭탄을 ArrayList에 넣어 생성 → move() 메소드 실행시켜 낙하

음식을 저장한 ArrayList의 index 값을 랜덤으로 지정 → 음식이 랜덤으로 내려오게 됨

```
// 음식 리스트에 넣기
int n = new Random().nextInt(20);
if (n == 0) {
    foodList.add(new Item(foodarr.get(random.nextInt(7)), width, height)); // 음식 랜덤으로 떨어지도록 함
}
for (int i = foodList.size() - 1; i >= 0; i--) {
    Item f = foodList.get(i);
    f.move();
    if (f.isDead)
        foodList.remove(i);
}
```

- checkCollision: 충돌 여부 확인

하트가 전부 사라지면 스레드 종료 & 게임오버 페이지 호출
& username과 최종 점수(finalscore)를 인자로 전달

```
// 게임 오버 - 생명 0
if (heart == 0) {
    finalscore = score; // 최종 점수
    System.out.println(finalscore);
    try {
        clip.stop(); // 음악 스탑
        adddb();
        dispose();
        new GameOverPage(finalscore);
    } catch (FontFormatException | IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```


<주요 함수>

4. actionPerformed: 일시정지 버튼 액션리스너

- 버튼을 누르면 스레드가 중지(suspend)
- 계속하기 버튼을 누르면 재시작(resume)
- 나가기 버튼을 누르면 게임 창 닫힘

5. GameThread: 음식, 폭탄이 내려가는 스레드

- run(): sleep(20) → 낙하 속도 지정
- 숫자가 작을 수록 속도 상승

```
// 일시정지 -----  
@Override  
public void actionPerformed(ActionEvent e) {  
    if (e.getSource() == button) {  
        clip.stop();  
        gThread.suspend(); // || 버튼을 누르면 Thread가 suspend되고 일시정지 팝업이 뜬.  
        stop.setVisible(true);  
    } else if (e.getSource() == bt1) {  
        stop.setVisible(false); // 계속하기 버튼을 누르면 일시정지 팝업이 사라지고 Thread가 resume됨  
        clip.start();  
        gThread.resume();  
        button.setFocusable(false);  
    } else if (e.getSource() == bt2) {  
        System.exit(0); // 나가기 버튼을 누르면 창이 종료됨.  
    }  
}
```

```
// 게임 스레드 -----  
class GameThread extends Thread {  
    int speed = 20; // 떨어지는 속도  
  
    @Override  
    public void run() { // 음식이 떨어지는 부분  
        while (true) {  
            panel.repaint(); // 화면 갱신  
            panel.makeFood();  
            try {  
                panel.makeBomb();  
            } catch (FontFormatException | IOException e1) {  
                // TODO Auto-generated catch block  
                e1.printStackTrace();  
            }  
            panel.move();  
            panel.checkCollision();  
            try {  
                sleep(speed);  
            } catch (InterruptedException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```


세부적인 기능 소개

본 게임 - Falling.java

<주요 함수>

6. **Item** : 음식, 폭탄에 관한 설정이 담긴 내부 클래스

- Random 함수를 통해 x좌표를 랜덤 지정

```
x = new Random().nextInt(width - 2 * w) + w;
```

- **move()**: y값을 일정하게 증가시켜 내려가는 효과 구현

```
void move() { // 음식이 떨어지는 효과
    y += 5;
    if (y > height + h) {
        isDead = true;
    }
}
```

7. **adddb** : 데이터베이스에 이름, 최종 점수 저장

```
public void adddb() throws FontFormatException, IOException {
    Main m = new Main();
    Start s = new Start();
    Ranking r = new Ranking(m.statement);

    System.out.println("유저이름 : " + s.user);
    System.out.println("최종점수 : " + finalscore);
    r.scoreadd(s.user,finalscore);
    r.dbconnection();
}
```

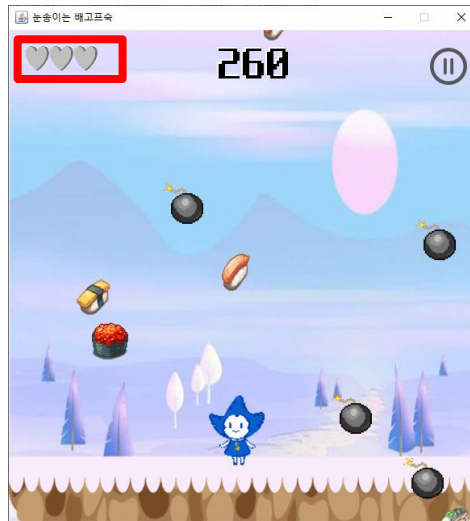
LEVEL

3

세부적인 기능 소개

게임 오버 및 랭킹 - GameOverPage & Ranking

1. 하트가 다 소진되어 게임이 끝나면 게임오버 페이지로 넘어감.
2. GameOverPage에서는 상단에 게임 플레이 점수를 표시, Ranking에서 가져온 이름과 점수의 list를 받아서 랭킹을 출력. (이때, 랭킹 5위까지 정보 출력)
3. 다시하기 버튼을 누르면 초기화면으로 이동
4. 나가기 버튼을 누르면 게임 종료



LEVEL

3

세부적인 기능 소개

게임 오버 및 랭킹 - GameOverPage & Ranking

<주요 함수>

GameOverPage.java

Ranking에서 이름과 점수를 가져와 상위 5개 출력

```
Ranking r = new Ranking(mp.statement);

String username[] = {"", "", "", "", ""};
String userscore[] = {"", "", "", "", ""};

for (int i=0; i<r.namelist.size(); i++) {
    if (i == 5) {
        break;
    }
    username[i] = ("" + r.namelist.get(i));
    userscore[i] = ("" + r.scorelist.get(i));
}
```

세부적인 기능 소개

게임 오버 및 랭킹 - GameOverPage & Ranking

<주요 함수>

Rankng.java

1. **scoreadd()**: Start에서 전달받은 이름과, Falling에서 전달받은 점수를 DB에 저장

```
String sql = "INSERT INTO `finalscore` (`name`, `score`)"  
            + "VALUES ('" + newname + "', " + newscore + ")";  
st.executeUpdate(sql);
```

2. **dbconnection()**: DB에 저장된 이름과 점수를 ArrayList에 저장

```
namelist = new ArrayList();  
scorelist = new ArrayList();  
  
try{  
    String sql = "select * "  
                + "from `finalscore`"  
                + "order by `score` DESC";  
    ResultSet rs = st.executeQuery(sql);  
  
    while(rs.next()){  
        name = rs.getString("name");  
        score = rs.getInt("score");  
        namelist.add(name);  
        scorelist.add(score);  
    }  
}
```

LEVEL

4

프로젝트에서 맡은 역할 및 기여도

눈송이는 배고프숙

김민지(1914386):

- 폰트 설정
- 시작 팝업 (Start.java)
- Falling.java에서 눈송이 움직이고 음식 떨어지는 메소드 및 스레드(move, makeFood, checkCollision, Item, GameThread)

김민지(2016680):

- 시작 페이지 (MainPage.java)
- Falling.java에서 폭탄, 배경음악 (makeBomb, loadAudio)
- 게임 오버 페이지 랭킹표

* 오디오 출처

♪ Music provided by 브금저장소 [BGM Storage]
♪ Track: 레트로 게임 - <https://youtu.be/Qp8HdY2U1B0>

노현진:

- 음식 랜덤하게 바뀌는 부분
- 데이터베이스 연동
- GameOverPage
- 페이지 레이아웃

오예지:

- 다시 시작 팝업 (Stop.java)
- Falling.java에서 눈송이 움직이기 (방향키 받아오기, 팝업창과 연결시켜 스레드, 방향키 정지/재생 등)

*매주 팀 회의를 진행하였으며, 팀원들 모두가 협력하여 프로젝트를 완성하였습니다.



감사합니다