

# 객체지향언어2 [8]

## 미니 프로젝트 결과 보고서

2191002 이서연

담당 교수명: 황기태 교수님

2022.12.13.

## 1. 작품 주제

‘닭을 키워라!’라는 게임은 아이템들을 이용하여 달걀에서 닭으로 키우는 게임이다. 아이템들은 내려오는 단어를 맞추면 얻을 수 있다.

## 2. 작품 개요

첫 화면에서는 자신의 이름을 입력하는 칸, 난이도 선택, 게임 시작 버튼, 단어 추가 버튼, 랭킹 보기 버튼이 있다. 자신의 이름을 입력하는 칸에 아무 입력도 하지 않으면 이름은 ‘누굴까’로 설정된다. 난이도를 선택하면 난이도에 따라 주어지는 제한시간, 단어 생성과 떨어지는 속도가 어려울수록 어렵게 달라진다. 단어 추가 버튼은 단어를 추가할 수 있는 버튼이다. 랭킹 보기 버튼은 사용자들의 점수를 보여주는 버튼이다.

자신의 이름을 입력하고 난이도 선택 후 게임 시작 버튼을 누르면 게임시작하는 화면으로 이동하게 된다. 게임시작 화면의 메뉴바에 있는 start 버튼을 누르면 타이머, 단어 생성, 노래가 시작된다. 이때 pause 버튼을 누르면 단어 생성 및 떨어지는 단어가 멈추게 되지만 타이머는 멈추지 않는다. 다시 단어들을 움직이게 하고 싶어 start버튼을 누르면 멈춰있던 단어들은 다시 움직이기 시작한다.

이 게임의 스테이지는 총 3개이다. 모든 스테이지에는 제한시간이 있고 다음 스테이지로 넘어갈수록 제한시간의 길이는 10씩 줄어들고 아이템의 속도는 빨라진다. 첫 스테이지는 달걀일 때 부화를 시키는 스테이지다. 달걀을 부화시키기 위해서는 온도는 35~39도가 적당하고 40도를 절대 넘기지 말아야 한다. 떨어지는 온도 아이템 중에서는 5도나 10도를 얻을 수 있게 하고 만약 얻은 온도 아이템이 40도가 넘게 된다면 게임은 끝난다. 또 부화시키기 위한 조건 중의 하나가 습도이다. 습도는 50~60%가 적정 습도이다. 습도 아이템은 10%를 얻을 수 있게 하고 얻은 습도 아이템이 60%가 넘게 된다면 게임은 끝난다. 제한시간 내에 온도, 습도의 조건을 모두 만족하게 한다면 다음 스테이지로 넘어가게 된다.

두 번째 스테이지는 병아리 키우는 스테이지다. 병아리 키우기 위해서는 먹이, 물, 온도가 필요하므로 먹이 아이템, 물 아이템, 온도 아이템을 얻어야 병아리를 키울 수 있다. 병아리가 필요한 먹이, 물을 100, 온도를 20이라고 설정해 놓고 먹이, 물이 100을 넘어가면 병아리가 과식으로 죽어 게임이 종료되고 온도가 20이 넘어가도 게임이 종료된다. 먹이, 물은 각각 10씩, 온도는 5나 10씩 얻을 수 있다. 제한시간 내에 먹이, 물, 온도의 조건을 모두 만족하게 한다면 다음 스테이지로 넘어가게 된다.

세 번째 스테이지는 닭을 지키는 스테이지다. 닭을 지키기 위해서는 울타리가 필요하므로 나무 아이템을 얻어야 한다. 나무 아이템은 5씩 떨어지고 30을 얻어야 닭을 지키는 울타리가 완성되어 모든 스테이지가 끝나게 된다.

모든 스테이지에는 시간 아이템이 있다. 이 아이템을 얻으면 시간은 3씩 늘어나게 된다. 세 번째 스테이지에서는 시간 아이템을 얻어도 시간은 늘어나지 않는다.

해당 스테이지에 조건이 맞는 아이템을 입력하여 맞추면 경쾌한 소리가 나오고 해당 스테이지에 조건이 맞지 않는 아이템을 입력하면 삑-하는 소리가 나온다.

첫 번째, 두 번째, 세 번째 스테이지에서 떨어지는 아이템들은 모두 같다. 그러므로 해당 스테이지에서 필요 없는 아이템을 얻게 된다면 기회가 1번 없어진다. 기회는 총 3번으로 3번의 기회를 다 쓰게 되면 게임이 종료된다.

스테이지가 끝나는 경우는 총 3가지이다. 첫 번째는 주어진 시간이 끝났을 경우, 두 번째는 목숨을 다 잃었을 경우, 세 번째는 각 스테이지마다 주어진 점수 조건에 벗어났을 경우이

다.

게임이 끝나면 자동으로 자신의 이름, 게임 끝난 스테이지, 게임 실행 시간이 저장된다.

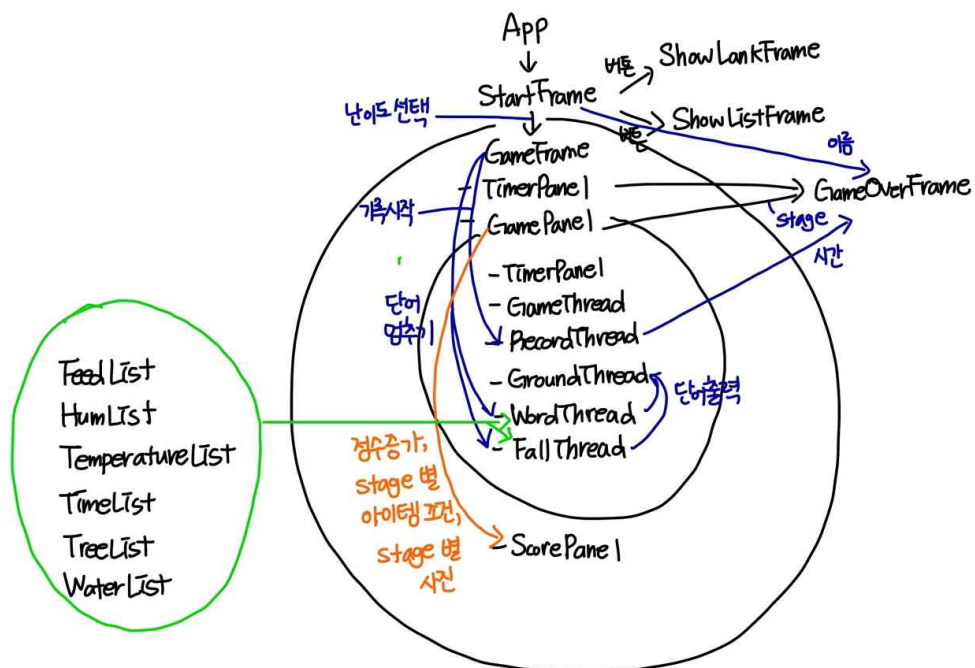
<아이템>

목숨 - 5개

Stage 1,2	-온도 (5씩) -습도 (10씩) -먹이 (10씩) -물 (5씩) -시간 (3씩)
Stage 3	-나무 (5씩)

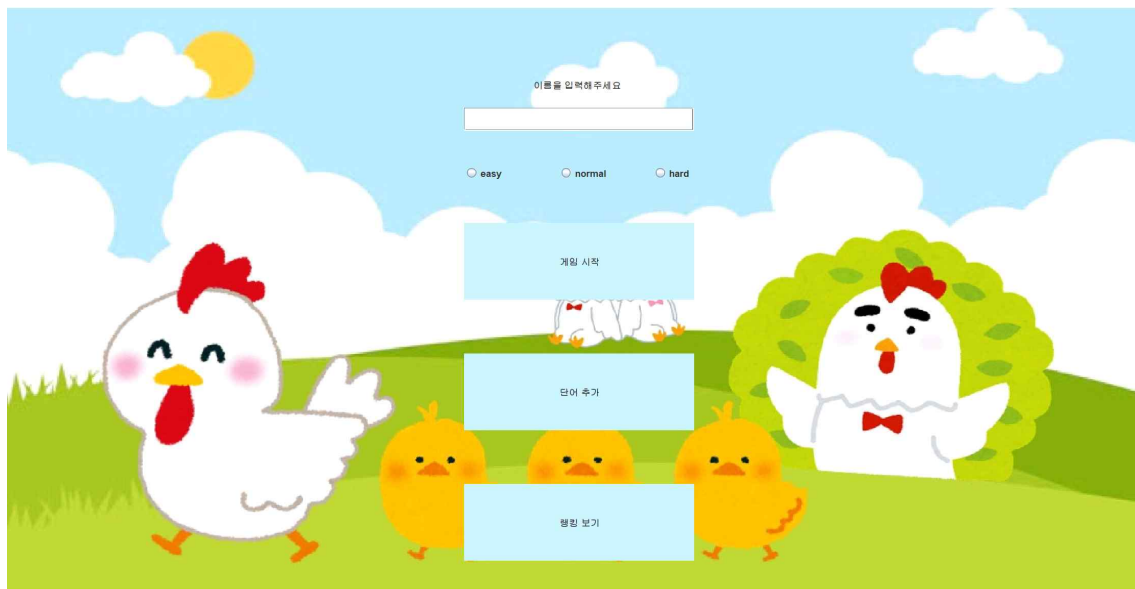
### 3. 프로그램 구조

- App
- GameFrame
- GameOverFrame
- ShowLankFrame
- ShowListFrame
- StartFrame
- StartPanel
- FeedList
- HumList
- TemperatureList
- TimeList
- TreeList
- WaterList
- GamePanel
- GameThread
- RecordThread
- GroundPanel
- WordThread
- FallThread
- ScorePanel
- TimerPanel
- TimerThread

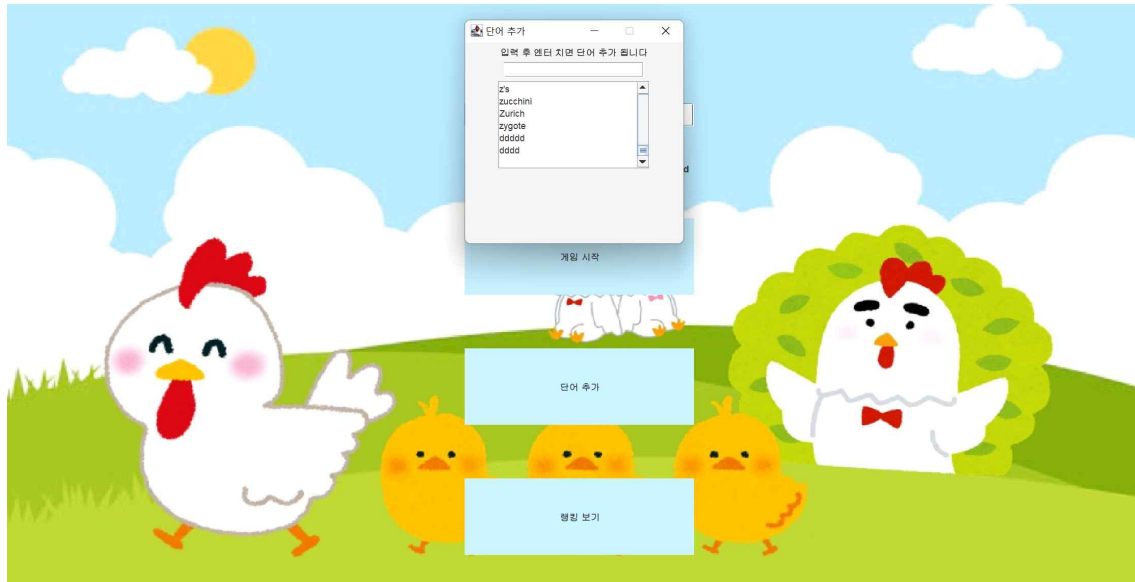


#### 4. 프로그램 실행 과정

##### 4-1) 첫 화면



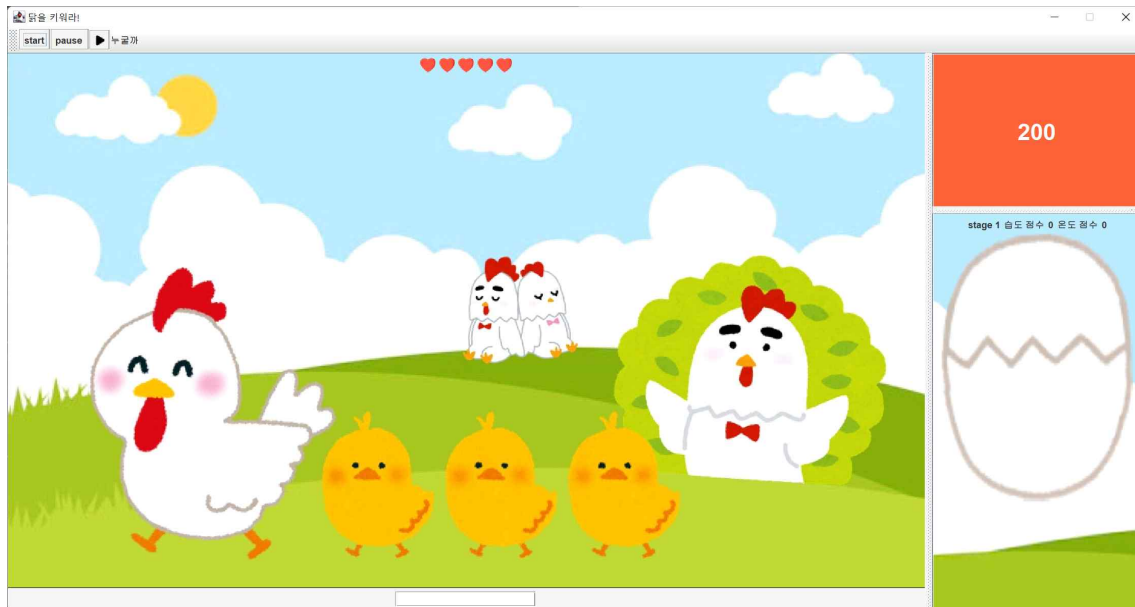
#### 4-2) 단어 추가 버튼 눌렀을 때



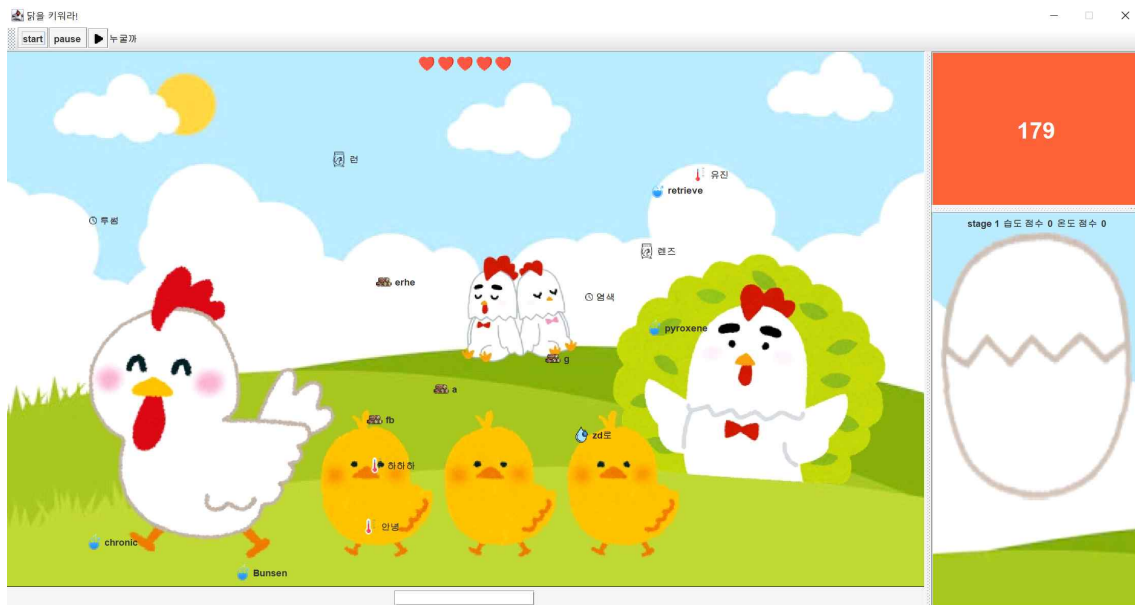
#### 4-3) 랭킹 보기 버튼을 눌렀을 때



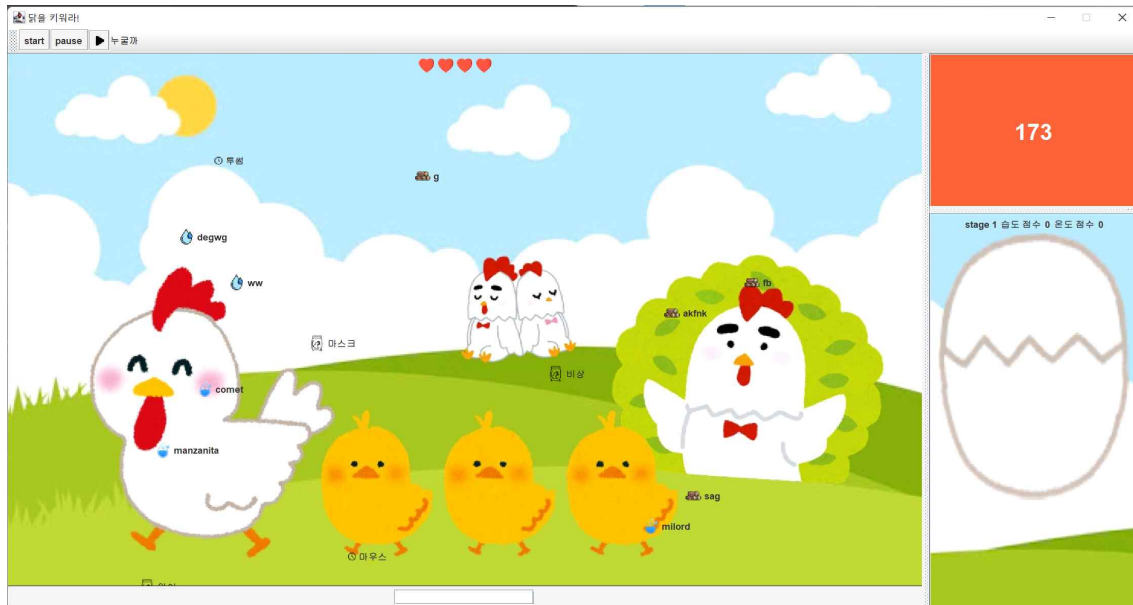
4-4) 이름, 난이도 선택 후 게임 시작 버튼을 눌렀을 경우  
(이름을 입력하지 않은 경우 - 누굴까로 설정)



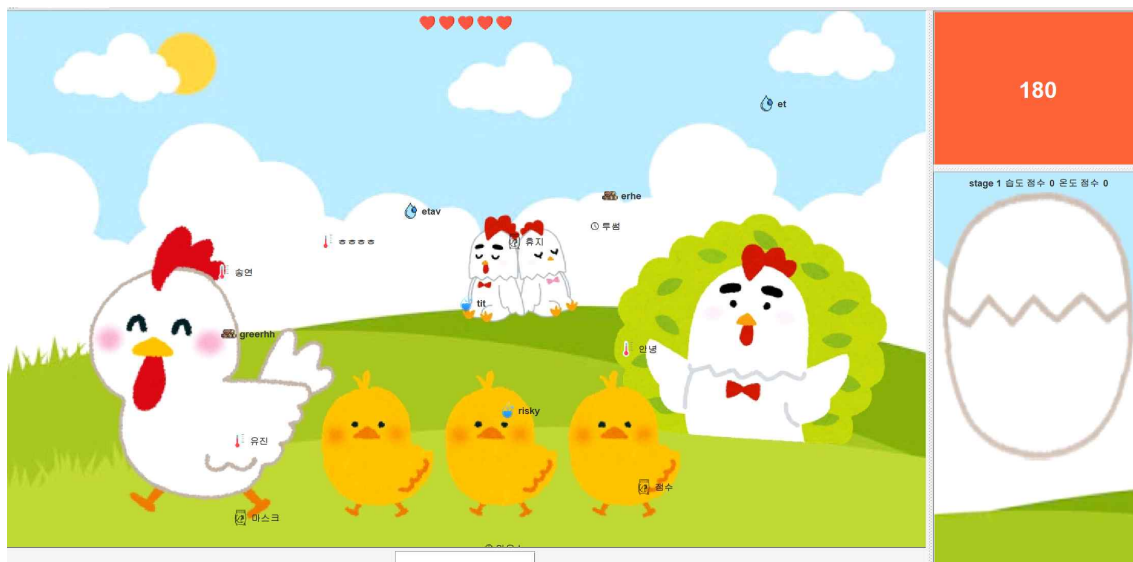
4-5) start 버튼을 누른 경우 ( 단어 생성, 단어 떨어짐, 타이머, 노래가 시작된다)



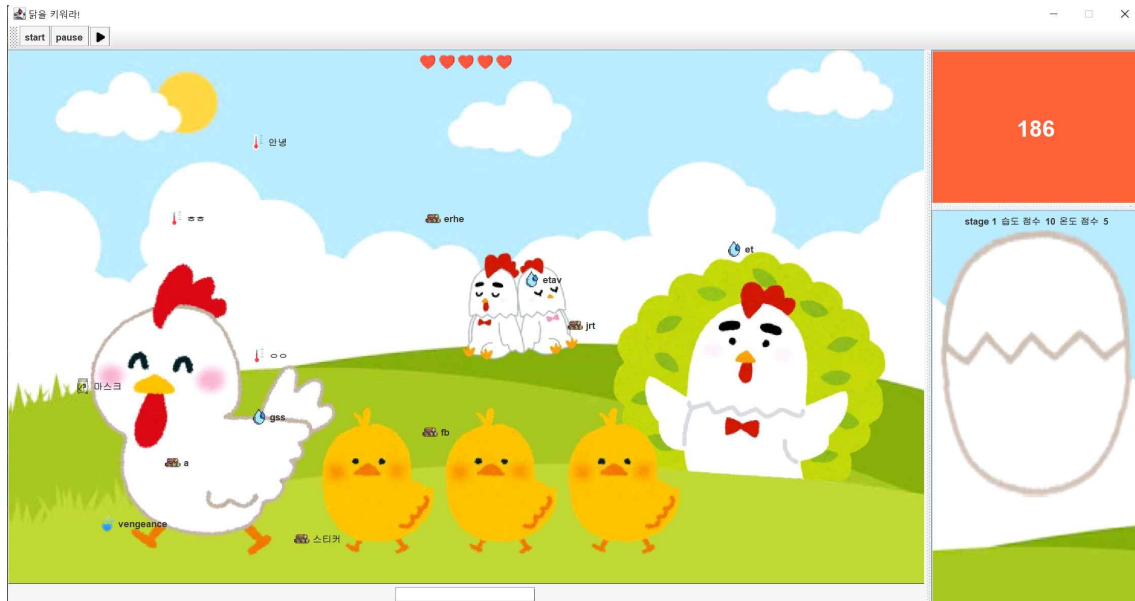
4-6) stage1 조건에 맞지 않는 단어를 입력한 경우 (목숨 -1)



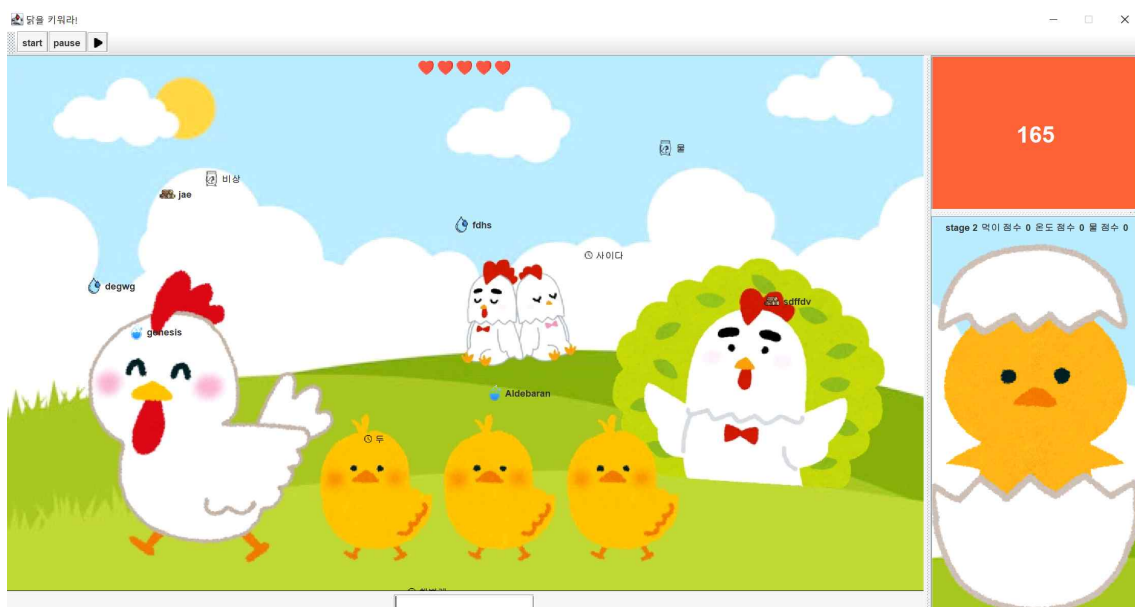
4-7) 시간 단어를 입력한 경우 (시간 증가)



4-7) stage1에서 습도,온도 단어를 각각 하나씩 맞춘 경우

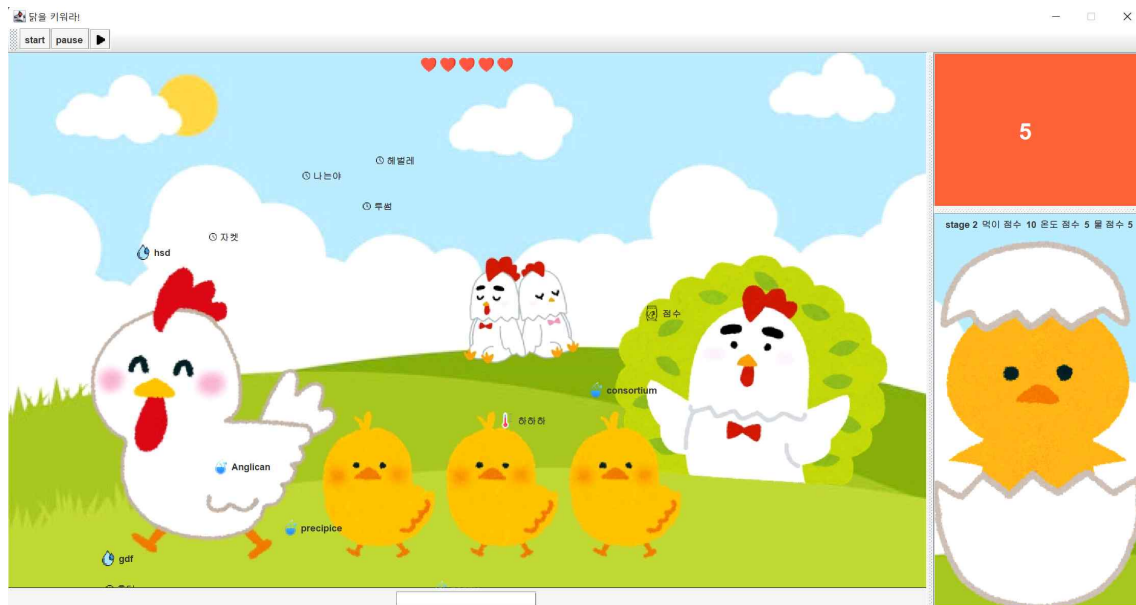


4-8) stage2 화면 (시간이 -10 된다)

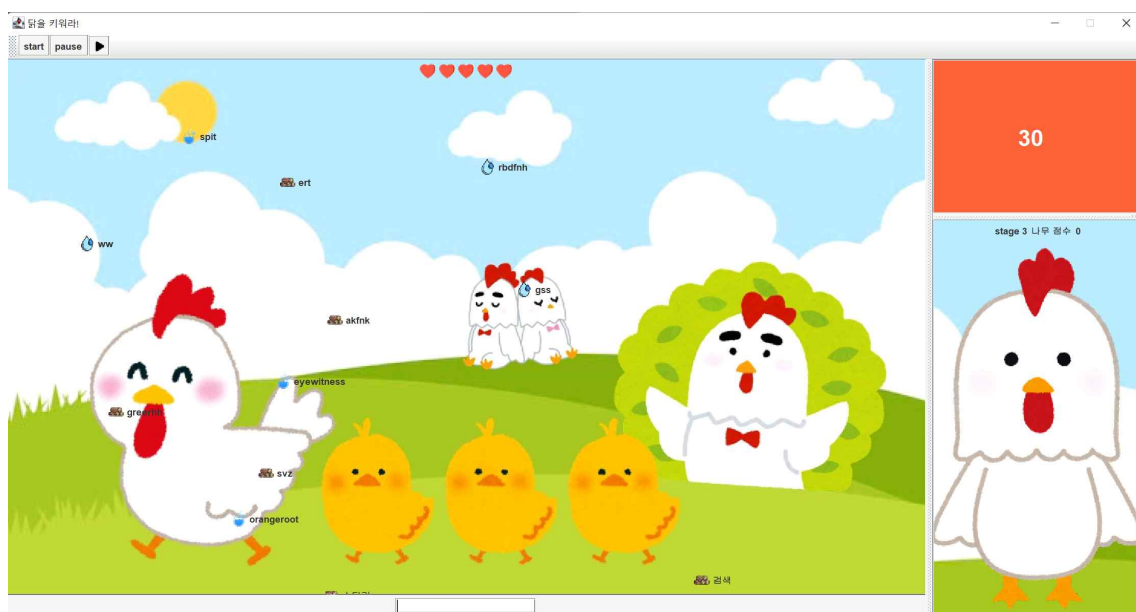




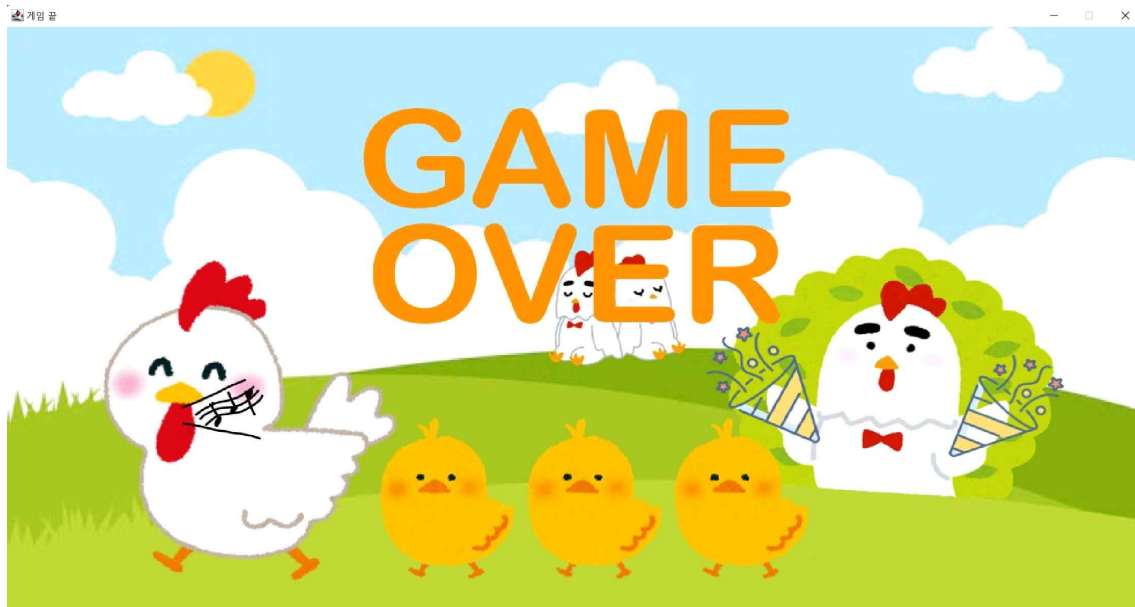
4-9) stage2에서 먹이, 온도, 물 단어 하나씩 맞출 경우



4-10) stage3 화면 (시간 -10)



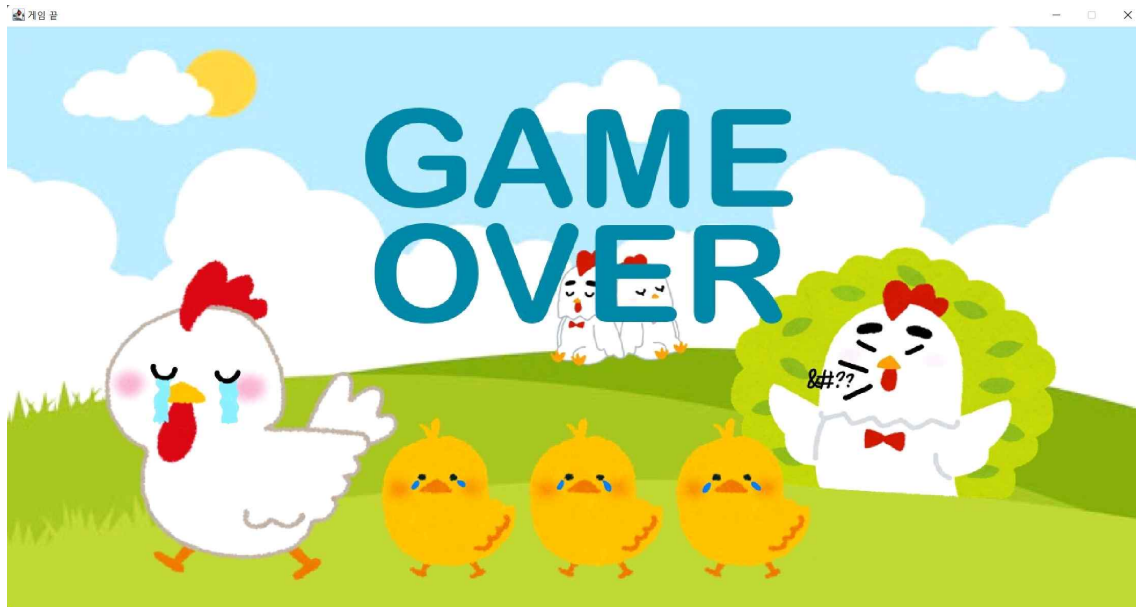
4-11) 모든 stage를 끝냈을 경우



4-12) 제한 시간으로 인해 게임이 종료되었을 경우



4-13) 목숨을 다 잃거나 스테이지 점수 조건에 어긋나 종료되었을 경우



## 5. 프로그램 소스 코드

### 5-1) App.java - 프로그램 시작

```
import frame.StartFrame;

public class App {

    public static void main(String[] args) {
        new StartFrame();
    }

}
```

## 5-2) GameFrame.java - 게임 시작 버튼 누르면 나오는 프레임

```
package frame;

import java.awt.BorderLayout;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JSplitPane;
import javax.swing.JToolBar;

import panel.GamePanel;
import panel.ScorePanel;
import panel.TimerPanel;

public class GameFrame extends JFrame {
    private JButton startBtn = new JButton("start ");
    private JButton pauseBtn = new JButton("pause ");

    private TimerPanel timerPanel = new TimerPanel();
    private GamePanel gamePanel = new GamePanel(timerPanel);
    private ScorePanel scorePanel = new ScorePanel();

    private File Clap = new File("background.wav");
    private int count = 0;
    private ImageIcon musicPlayBtn = new ImageIcon("play.png");
    private ImageIcon musicPauseBtn = new ImageIcon("pause.png");

    public GameFrame() {
        super("답을 키워라!");
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocation(100, 50);
        setSize(1500, 800);
        makeToolBar();
        makeSplitPane();
        setVisible(true);
    }
}
```

```

//메뉴바 만드는 함수
private void makeToolBar() {

    JToolBar tBar = new JToolBar();
    tBar.add(startBtn);
    startBtn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            if (count == 0) {
                PlaySound(Clap, "play");
                gamePanel.startWordThread();
                count++;
            } else {
                if (gamePanel.getStop() == true) {
                    gamePanel.resume();
                }
            }
            timerPanel.runTimer(); // 시작하기 버튼 누르면
            gamePanel.startRecord();

        }
    });
    tBar.add(pauseBtn);
    pauseBtn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (gamePanel.getStop() == false) {
                gamePanel.stop();
            }
        }
    });

    JButton playBtn = new JButton(musicPlayBtn);
    tBar.add(playBtn);
    playBtn.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (playBtn.getIcon() == musicPlayBtn) {
                playBtn.setIcon(musicPauseBtn);
                PlaySound(Clap, "pause");
            } else {
                playBtn.setIcon(musicPlayBtn);
                PlaySound(Clap, "play");
            }
        }
    });
    tBar.add(new JLabel(StartFrame.userName));
    getContentPane().add(tBar, BorderLayout.NORTH);
}

//패널 구간 나누기
private void makeSplitPane() {
    JSplitPane hPane = new JSplitPane();
    hPane.setOrientation(JSplitPane.HORIZONTAL_SPLIT); // 수평 분할
    hPane.setDividerLocation(1200); // 선의 초기 위치 설정
    getContentPane().add(hPane, BorderLayout.CENTER);

    JSplitPane vPane = new JSplitPane();
    vPane.setOrientation(JSplitPane.VERTICAL_SPLIT);
    vPane.setDividerLocation(200);

    hPane.setRightComponent(vPane);
    hPane.setLeftComponent(gamePanel);
    vPane.setTopComponent(timerPanel);
    vPane.setBottomComponent(scorePanel);
}

```

타이머 시작

```
//배경음악 실행하는 함수
public void PlaySound(File Sound, String action) {
    try {
        Clip clip = AudioSystem.getClip();
        clip.open(AudioSystem.getAudioInputStream(Sound));
        if (action.equals("play")) {
            clip.start();
            clip.loop(10);
        }
        if (action.equals("pause")) {
            clip.stop();
        }
    } catch (Exception e) {
    }
}
```

### 5-3) GameOverFrame.java - 게임 종료하였을 때 나오는 프레임

```
import java.awt.Graphics;
import java.awt.Image;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JPanel;

public class GameOverFrame extends JFrame {
    private String user = StartFrame.userName;
    private ImageIcon lose = new ImageIcon("lose.jpg");
    private ImageIcon win = new ImageIcon("win.jpg");
    private ImageIcon timeOver = new ImageIcon("timeover.jpg");
    private Image img;
    public GameOverFrame(String result,int stage,int record) {
        super("게임 끝");
        setResizable(false);
        setSize(1500,800);
        setLocation(100,50);
        setContentPane(new GameOverPanel());
        setVisible(true);
        if(result.equals("failure")){
            System.out.println("실패");
            img = lose.getImage();
        }
        if(result.equals("win")){
            System.out.println("성공");
            img = win.getImage();
        }
        if(result.equals("timeover")) {
            System.out.println("타임오버");
            img = timeOver.getImage();
        }
        //랭킹 만들기
        try(FileWriter fw = new FileWriter("lank.txt", true);
            BufferedWriter bw = new BufferedWriter(fw);
            PrintWriter out = new PrintWriter(bw))
        {
            out.write(user+" "+stage+" "+record+"\n");
        }
        catch (IOException e1) {
        }
    }
    class GameOverPanel extends JPanel{
        @Override
        public void paintComponent(Graphics g) {
            g.drawImage(img, 0, 0,getWidth(),getHeight(), null);
        }
    }
}
```

### 5-4) ShowLankFrame.java - 랭킹 보기 버튼 누르면 나오는 프레임

```

package frame;

import java.awt.Font;
import java.awt.Graphics;
import java.awt.Image;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;

import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class ShowLankFrame extends JFrame {
    public ShowLankFrame() {
        super("점수 보기");
        setContentPane(new ShowLankPanel());
        setSize(1500,800);
        setLocation(100,50);
        setLayout(null);
        setVisible(true);
    }
    class ShowLankPanel extends JPanel {
        private ImageIcon icon = new ImageIcon("lank.jpg");
        private Image img = icon.getImage();
        private String[] list;
        private int y=0;
        public ShowLankPanel() {
            setResizable(false);
            try {
                Scanner sc = new Scanner(new
FileReader("lank.txt"));
                while(sc.hasNext()) {
                    String word = sc.nextLine();
                    list = word.split(" ");
                    JLabel nameLabel= new JLabel(list[0]);
                    JLabel stageLabel= new JLabel(list[1]);
                    JLabel timeLabel = new JLabel(list[2]);
                    nameLabel.setFont(new Font("HY견고딕
보통",Font.BOLD,30));

                    nameLabel.setLocation(460,130+y);
                    nameLabel.setSize(100,100);
                    nameLabel.setVisible(true);

                    stageLabel.setFont(new Font("HY견고딕
보통",Font.BOLD,30));

                    stageLabel.setLocation(750,130+y);
                    stageLabel.setSize(100,100);
                    stageLabel.setVisible(true);

                    timeLabel.setFont(new Font("HY견고딕
보통",Font.BOLD,30));

                    timeLabel.setLocation(950,130+y);
                    timeLabel.setSize(100,100);
                    timeLabel.setVisible(true);
                    y=y+50;
                    add(nameLabel);
                    add(stageLabel);
                    add(timeLabel);
                }
                sc.close();
            } catch (FileNotFoundException e) {
                e.printStackTrace();
            }
        }
        @Override
        public void paintComponent(Graphics g) {
            g.drawImage(img, 0, 0,getWidth(),getHeight(), null);
        }
    }
}

```



5-5) ShowListFrame.java - 단어 추가 버튼 누르면 나오는 프레임

```

package frame;

import java.awt.Container;
import java.awt.FlowLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedWriter;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class ShowListFrame extends JFrame {
    private JTextField word = new JTextField(20);
    private JTextArea wordList = new JTextArea(7,20);
    public ShowListFrame() {
        super("단어 추가");
        setResizable(false);
        Container c = getContentPane();
        c.setLayout(new FlowLayout());
        c.add(new JLabel("입력 후 엔터 치면 단어 추가 됩니다"));
        c.add(word);
        c.add(new JScrollPane(wordList));

        //wordList에 word.txt 쓰기
        try {
            Scanner sc = new Scanner(new
FileReader("humWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordList.append(word+'\n');
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }
}

```

```

//단어입력하고 엔터 칠 경우 단어 추가
word.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JTextField t = (JTextField)e.getSource();
        //아무것도 안 쳤을 경우에 추가 되지 않도록
        if(!t.getText().equals(""))
        {
            try(FileWriter fw = new
FileWriter("humWords.txt", true);
BufferedWriter bw = new
BufferedWriter(fw);
PrintWriter out = new
PrintWriter(bw))
            {
                out.write(t.getText()+"\n");
            }
            catch (IOException e1) {
            }
            wordList.append(t.getText()+"\n"); //목록에
추가
            t.setText("");
        }
    }
});
setLocation(700,100);
setSize(300,300);
setVisible(true);
}
}

```

```

package frame;
import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.ButtonGroup;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextField;

public class StartFrame extends JFrame {
    private JLabel msg;
    private JTextField name;
    private JRadioButton level [];
    private String levelList[] = {"easy","normal","hard"};
    private JButton startBtn;
    private JButton addWordBtn;
    private JButton lankBtn;
    private ImageIcon icon = new ImageIcon("background.jpg");
    private Image img = icon.getImage();
    private StartPanel panel = new StartPanel();

    public static int setLevel;
    public static String userName;

    public StartFrame() {
        super("답을 키워라!");
        setResizable(false);
        setSize(1500,800);
        setLocation(100,50);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setContentPane(panel);
        setVisible(true);
    }

    class StartPanel extends JPanel{
        @Override
        public void paintComponent(Graphics g) {
            g.drawImage(img, 0, 0,getWidth(),getHeight(), null);
        }
        public StartPanel() {
            setLayout(null);
            msg = new JLabel("이름을 입력해주세요");
            msg.setLocation(690, 50);
            msg.setSize(200,100);
            add(msg);

            name = new JTextField(20);
            name.setLocation(600,130);
            name.setSize(300,30);
            add(name);
            name.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    if(name.getText().equals(""))
                        userName="누굴까";
                    else
                        userName=name.getText();
                }
            });
        }
    }
}

```

```

ButtonGroup group = new ButtonGroup();
    level = new JRadioButton[3];
    for(int i=0;i<level.length;i++) {
        level[i]=new JRadioButton(levelList[i]);
        level[i].setLocation(600+i*123, 190);
        level[i].setSize(80,50);
        level[i].setOpaque(false);
        group.add(level[i]);
        add(level[i]);
        level[i].addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent
e) {

                String s = e.getActionCommand();
                if(s.equals(level[0].getText())) {
                    setLevel=1;
                }
                else if(s.equals(level[1].getText()))
{
                    setLevel=2;
                }
                else if(s.equals(level[2].getText()))
{
                    setLevel=3;
                }
            }
        });
    }

    startBtn = new JButton("게임 시작");
    startBtn.setSize(300, 100);
    startBtn.setLocation(600,280);
    startBtn.setBackground(new Color(198,239,255));
    startBtn.setBorderPainted(false);
    add(startBtn);

    addWordBtn = new JButton("단어 추가");
    addWordBtn.setSize(300,100);
    addWordBtn.setLocation(600,450);
    addWordBtn.setBackground(new Color(198,239,255));
    addWordBtn.setBorderPainted(false);
    add(addWordBtn);
    setVisible(true);

    lankBtn = new JButton("랭킹 보기");
    lankBtn.setSize(300,100);
    lankBtn.setLocation(600,620);
    lankBtn.setBackground(new Color(198,239,255));
    lankBtn.setBorderPainted(false);
    add(lankBtn);
    setVisible(true);
}
}

```

```

startBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        setVisible(false);
        new GameFrame();
        System.out.println("startFrame:"+setLevel);
        System.out.println(userName);
    }
});

addWordBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new ShowListFrame();
    }
});

lankBtn.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        new ShowLankFrame();
    }
});
    }
}

```

```

package panel;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.Image;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.File;
import java.util.HashMap;
import java.util.Vector;

import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JTextField;

import frame.GameOverFrame;
import frame.StartFrame;
import list.FeedList;
import list.HumList;
import list.TemperatureList;
import list.TimeList;
import list.TreeList;
import list.WaterList;

public class GamePanel extends JPanel {
    private ImageIcon icon = new ImageIcon("background.jpg");
    private Image img = icon.getImage();

    private ScorePanel scorePanel = new ScorePanel();
    private TimerPanel timerPanel;

    private JPanel textAreaPanel;
    private JTextField inputField = new JTextField(20);

    private HumList humList = new HumList(); // 단어리스트 가져오기
    private TemperatureList temList = new TemperatureList();
    private WaterList waterList = new WaterList();
    private TreeList treeList = new TreeList();
    private TimeList timeList = new TimeList();
    private FeedList feedList = new FeedList();

    private HashMap<JLabel, Integer> wordHash = new HashMap<JLabel, Integer>();
    private Vector<JLabel> fallVector = new Vector<JLabel>();
    private ImageIcon[] images = { new ImageIcon("heart.png"), // [0]
        new ImageIcon("heart.png"), // [1]
        new ImageIcon("heart.png"), // [2]
        new ImageIcon("heart.png"), // [3]
        new ImageIcon("heart.png") // [4]
    };
    private JLabel[] imageLabel = new JLabel[5];
    private int heartLength = 4;

    private File Correct = new File("correct.wav");
    private File Error = new File("error.wav");

    private int stage = 1;
    private int delay;
    private int fallDelay;
    private GroundPanel groundPanel = new GroundPanel();
    private GameThread gameThread = new GameThread();
    private RecordThread recordThread = new RecordThread();
    private boolean gameOverFlag = false;

    private int record=0;

```

```

public GamePanel(TimerPanel timerPanel) {
    this.timerPanel=timerPanel;
    setLayout(new BorderLayout());
    add(groundPanel, BorderLayout.CENTER);

    setBackground(Color.black);
    textAreaPanel = new JPanel(); // 단어 쓰는 바탕
    // textArea.setBackground(Color.GRAY);
    textAreaPanel.add(inputField); // 단어 쓰는 곳 넣기
    changeCondition(inputField);
    add(textAreaPanel, BorderLayout.SOUTH);

    gameThread.start();

}

public void startRecord() {
    recordThread.start();
}

public void setSpeed() {
    if (StartFrame.setLevel == 1) {
        delay = 800;
    } else if (StartFrame.setLevel == 2) {
        delay = 900;
    } else if (StartFrame.setLevel == 3) {
        delay = 1000;
    }
}

public void setFallSpeed() {
    if (StartFrame.setLevel == 1) {
        fallDelay = 15;
    } else if (StartFrame.setLevel == 2) {
        fallDelay = 20;
    } else if (StartFrame.setLevel == 3) {
        fallDelay = 40;
    }
}

public void startWordThread() {
    groundPanel.startFallWord();
}

public boolean getStop() {
    return groundPanel.getStopFlag();
}

public void stop() {
    groundPanel.stopFallWord();
}

public void resume() {
    groundPanel.wordTh.resumeWordMove();
    groundPanel.fallTh.resumeFallMove();
}

```



```

        public void changeCondition(JTextField inputField) {
            inputField.addActionListener(new ActionListener() {
                // 1-feed 2-hum 3-tem 4-time 5-tree 6-water
                @Override
                public void actionPerformed(ActionEvent e) {
                    JTextField t = (JTextField) e.getSource();
                    if (stage == 1) {
                        for (int i = 0; i < fallVector.size(); i++) {
                            JLabel label = fallVector.get(i);
                            if
(t.getText().equals(label.getText())) {
                                // 입력한 값이 습도단어랑
                                같은 경우
                                if (wordHash.get(label) ==
                                2) {
                                    fallVector.remove(label);
                                    groundPanel.remove(label);
                                    scorePanel.humIncrease();
                                    PlaySound(Correct);
                                }
                                else if
                                (wordHash.get(label) == 4) {
                                    fallVector.remove(label);
                                    groundPanel.remove(label);
                                    timerPanel.isIncreaseTimer();
                                    PlaySound(Correct);
                                }
                                // 입력한 값이 온도단어랑
                                else if
                                (wordHash.get(label) == 3) {
                                    fallVector.remove(label);
                                    groundPanel.remove(label);
                                    scorePanel.temIncrease();
                                    PlaySound(Correct);
                                }
                                } else if
                                (wordHash.get(label) == 1 || wordHash.get(label) == 5
                                ||
                                wordHash.get(label) == 6) {
                                    fallVector.remove(label);
                                    groundPanel.remove(label);
                                    if(!(heartLength<0))
                                    groundPanel.remove(imageLabel[heartLength]);
                                    heartLength--;
                                    PlaySound(Error);
                                }
                            }
                        }
                    }
                }
            }
        }

```

```

// 습도 점수, 온도 점수가 조건에 맞으면 다음 스테이지로 넘어감
scorePanel.getTemScore() == 10) {
    if (scorePanel.getHumScore() == 10 &&
        stage = 2;
        fallDelay+=10;
        timerPanel.isStageTimer();
        scorePanel.changeTwo();
    }
    if (scorePanel.getHumScore() > 10 ||
        recordThread.interrupt();
        new
GameOverFrame("failure",stage,record);
    }
    // 1-feed 2-hum 3-tem 4-time 5-tree
6-water
    } else if (stage == 2) {
        for (int i = 0; i < fallVector.size(); i++) {
            JLabel label = fallVector.get(i);
            if
(t.getText().equals(label.getText())) {
                // 입력한 값이 먹이단어랑
                같은 경우
                if (wordHash.get(label) ==
1) {
                    fallVector.remove(label);
                    groundPanel.remove(label);
                    scorePanel.feedIncrease();
                    PlaySound(Correct);
                }
                // 입력한 값이 물단어랑
                같은 경우
                if (wordHash.get(label) == 6) {
                    fallVector.remove(label);
                    groundPanel.remove(label);
                    scorePanel.waterIncrease();
                    PlaySound(Correct);
                }
                else if
(wordHash.get(label) == 3) {
                    fallVector.remove(label);
                    groundPanel.remove(label);
                    scorePanel.temIncrease();
                    PlaySound(Correct);
                }
                else if
(wordHash.get(label) == 4) {
                    fallVector.remove(label);
                    groundPanel.remove(label);
                    timerPanel.isIncreaseTimer();
                    PlaySound(Correct);
                }
            }
        }
    }
}

```

```

// 먹이,물,온도 단어 이외의 단어 입력 시 목숨 1개 제거
(wordHash.get(label) == 2 || wordHash.get(label) == 5) {
    fallVector.remove(label);
    groundPanel.remove(label);
    if(!(heartLength<0))
        groundPanel.remove(imageLabel[heartLength]);
    heartLength--;
    PlaySound(Error);
}
}
}
// 먹이,물,온도 단어가 조건에 맞을 시 다음
스테이지로 넘어감
if (scorePanel.getFeedScore() == 10 &&
    scorePanel.getWaterScore() == 10 &&
    scorePanel.getTemScore() == 10) {
    stage = 3;
    fallDelay+=10;
    timerPanel.isStageTimer();
    scorePanel.changeThree();
}
if (scorePanel.getFeedScore() > 10 ||
    scorePanel.getWaterScore() > 10 ||
    scorePanel.getTemScore() > 10) {
    recordThread.interrupt();
    new
    GameOverFrame("failure",stage,record);
} else if (stage == 3) {
    for (int i = 0; i < fallVector.size(); i++) {
        JLabel label = fallVector.get(i);
        if
        (t.getText().equals(label.getText())) {
            // 입력한 값이 나무단어랑
            같은 경우
            if
            (wordHash.get(fallVector.get(i)) == 5) {
                fallVector.remove(label);
                groundPanel.remove(label);
                scorePanel.treeIncrease();
                PlaySound(Correct);
            }
        }
    }
}

```

[illegible]

```

@Override
    public void paintComponent(Graphics g) {
        g.drawImage(img, 0, 0, null);
    }

    public void startFallWord() {
        wordTh.start();
        fallTh.start();
    }

    private boolean stopFlag = false;

    public boolean getStopFlag() {
        return stopFlag;
    }

    public void stopFallWord() {
        stopFlag = true;
    }

    //단어 생성 스레드
    class WordThread extends Thread {
        synchronized private void resumeWordMove() {
            stopFlag = false;
            this.notify();
        }
        synchronized private void waitWordFlag() { // wait
            try {
                this.wait(); // 무한 대기 .notify()가
            } catch (InterruptedException e) {
                e.printStackTrace();
            } // 기다리면서 중단 된 상태
        }
    }

    @Override
    public void run() {
        int x;
        JLabel pickWord;
        while (true) {
            pickWord = getPickWord();
            x = (int) (Math.random() *
                (groundPanel.getWidth() - 250));

            pickWord.setSize(200, 200);
            pickWord.setLocation(x, 0);
            // fallWord.setVisible(true);
            groundPanel.add(pickWord);
            repaint();
            try {
                setSpeed();
                if (stopFlag == true)
                    waitWordFlag();
                repaint();
                Thread.sleep(delay);
            } catch (InterruptedException e) {
                return;
            }
        }
    }
}

```

사용할때는 try-catch 사용 //멈추는 것을 할 때에는 무조건 flag.notify사용하기

불러질 때까지

```

//떨어지는 단어 스레드
class fallThread extends Thread {
    synchronized private void resumeFallMove() {
        stopFlag = false;
        this.notify();
    }
    synchronized private void waitFallFlag() { // wait
        //멈추는 것을 할 때에는 무조건 flag, notify 사용하기
        try {
            this.wait(); // 무한 대기 .notify()가
            //불러질 때까지
        } catch (InterruptedException e) {
            e.printStackTrace();
        } // 기다리면서 중단 된 상태
    }
    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(500);
                for (int i = 0; i < fallVector.size();
                    i++) {
                        JLabel fallWord =
                            fallVector.get(i);
                            setFallSpeed();
                            fallWord.setLocation(fallWord.getX(), fallWord.getY() + fallDelay);
                        }
                        if (stopFlag == true)
                            waitFallFlag();
                        repaint();
                    } catch (InterruptedException e) {
                        return;
                    }
                }
            }
        }
    }
}

```

```

//6종류의 Vector중에 하나를 골라 fallVector와 wordHash에 넣는 함수
public JLabel getPickWord() {

    int x;
    int num;
    int number;
    JLabel word = null;
    num = (int) ((Math.random() * 6) + 1);
    if (num == 1) {
        number = (int) (Math.random() *
(feedList.getFeedVector().size()));
        word = feedList.getFeedVector().get(number);
    } else if (num == 2) {
        number = (int) (Math.random() *
(humList.getHumVector().size()));
        word = humList.getHumVector().get(number);
    } else if (num == 3) {
        number = (int) (Math.random() *
(temList.getTemVector().size()));
        word = temList.getTemVector().get(number);
    } else if (num == 4) {
        number = (int) (Math.random() *
(timeList.getTimeVector().size()));
        word = timeList.getTimeVector().get(number);
    } else if (num == 5) {
        number = (int) (Math.random() *
(treeList.getTreeVector().size()));
        word = treeList.getTreeVector().get(number);
    } else if (num == 6) {
        number = (int) (Math.random() *
(waterList.getWaterVector().size()));
        word = waterList.getWaterVector().get(number);
    }
    wordHash.put(word, num);
    fallVector.add(word);
    return word;
}

public void PlaySound(File Sound) {
    try {
        Clip clip = AudioSystem.getClip();
        clip.open(AudioSystem.getAudioInputStream(Sound));
        clip.start();

    } catch (Exception e) {}
}
}

```

5-8) ScorePanel.java - 해당 스테이지, 아이템 점수, 스테이지별 이미지 보여주는  
패널



```

package panel;

import java.awt.Color;
import java.awt.Graphics;
import java.awt.Image;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class ScorePanel extends JPanel {
    private static int humScore = 0;
    private static int temScore = 0;
    private static int feedScore = 0;
    private static int waterScore = 0;
    private static int treeScore = 0;

    private static JLabel firstLabel;
    private static JLabel secondLabel;
    private static JLabel thirdLabel;

    private static JLabel humScoreLabel = new
JLabel(Integer.toString(humScore));
    private static JLabel temScoreLabel = new
JLabel(Integer.toString(temScore));
    private static JLabel feedScoreLabel = new
JLabel(Integer.toString(feedScore));
    private static JLabel waterScoreLabel = new
JLabel(Integer.toString(waterScore));
    private static JLabel treeScoreLabel = new
JLabel(Integer.toString(treeScore));

    private static JLabel stageLabel;
    private static ImageIcon stageOneImageIcon = new ImageIcon("egg.png");
    private static JLabel stageImage = new JLabel(stageOneImageIcon);
    private static ImageIcon stageTwoImageIcon = new ImageIcon("chick.png");
    private static ImageIcon stageThreeImageIcon = new
ImageIcon("chicken.png");

    private ImageIcon icon = new ImageIcon("back.png");
    private Image img = icon.getImage();
    @Override
    public void paintComponent(Graphics g) {
        g.drawImage(img, 0, 0, getWidth(), getHeight(), null);
    }
    public ScorePanel() {
        firstLabel = new JLabel("습도 점수");
        secondLabel = new JLabel("온도 점수");
        thirdLabel = new JLabel("물 점수");
        stageLabel = new JLabel("stage 1");
        add(stageLabel);
        add(firstLabel);
        add(humScoreLabel);
        add(feedScoreLabel);
        feedScoreLabel.setVisible(false);
        add(treeScoreLabel);
        treeScoreLabel.setVisible(false);
        add(secondLabel);
        add(temScoreLabel);
        add(thirdLabel);
        thirdLabel.setVisible(false);
        add(waterScoreLabel);
        waterScoreLabel.setVisible(false);
        add(stageImage);
    }
}

```

```

public void changeTwo() {
    System.out.println("2");
    temScore=0;
    humScoreLabel.setVisible(false);
    firstLabel.setText("먹이 점수");
    feedScoreLabel.setVisible(true);
    secondLabel.setText("온도 점수");
    temScoreLabel.setText("0");
    thirdLabel.setVisible(true);
    waterScoreLabel.setVisible(true);
    stageLabel.setText("stage 2");
    stageImage.setIcon(stageTwoImageIcon);
}
public void changeThree() {
    System.out.println("3");
    feedScoreLabel.setVisible(false);
    temScoreLabel.setVisible(false);
    waterScoreLabel.setVisible(false);
    firstLabel.setText("나무 점수");
    treeScoreLabel.setVisible(true);
    secondLabel.setVisible(false);
    thirdLabel.setVisible(false);
    stageLabel.setText("stage 3");
    stageImage.setIcon(stageThreeImageIcon);
}

public void humIncrease() {
    humScore += 10;
    humScoreLabel.setText(Integer.toString(humScore));
}
public void temIncrease() {
    temScore += 5;
    temScoreLabel.setText(Integer.toString(temScore));
}

public void feedIncrease() {
    feedScore += 10;
    feedScoreLabel.setText(Integer.toString(feedScore));
}

public void waterIncrease() {
    waterScore += 5;
    waterScoreLabel.setText(Integer.toString(waterScore));
}
public void treeIncrease() {
    treeScore += 5;
    treeScoreLabel.setText(Integer.toString(treeScore));
}

public int getHumScore() {return humScore;}
public int getTemScore() {return temScore;}
public int getFeedScore() {return feedScore;}
public int getWaterScore() {return waterScore;}
public int getTreeScore() {return treeScore;}
}

```

#### 5-9) TimerPanel.java - 타이머 보여주는 패널

```
package panel;

import java.awt.Color;
import java.awt.Font;
import java.awt.Graphics;
import java.awt.Image;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JPanel;

import frame.GameOverFrame;

public class TimerPanel extends JPanel {
    private JLabel timer = new JLabel("200");
    private int count = 200;
    private ImageIcon icon = new ImageIcon("clock.png");
    private Image img = icon.getImage();
    //
    // @Override
    // public void paintComponent(Graphics g) {
    //     g.drawImage(img, 0, 0, getWidth(), getHeight(), null);
    // }
    // public TimerPanel() {
    //     setLayout(null);
    //     timer.setFont(new Font("HY견고딕 보통",Font.BOLD,30));
    //     timer.setForeground(Color.white);
    //     timer.setSize(100,100);
    //     timer.setLocation(110,50);
    //     setBackground(new Color(255,96,55));
    //     add(timer);
    // }

    public void runTimer() {
        TimerThread th = new TimerThread(timer, 1000);
        th.start();
    }

    private boolean flag = false;
    private boolean deFlag = false;
    private boolean stageFlag = false;

    public void isIncreaseTimer() {
        flag = true;
    }

    public void isDecreaseTimer() {
        deFlag = true;
    }

    public void isStageTimer() {
        stageFlag = true;
    }
}
```

```

class TimerThread extends Thread {
    private JLabel text = null;

    private int delay = 0;

    public TimerThread(JLabel text, int delay) {
        this.text = text;
        this.delay = delay;
    }

    @Override
    public void run() {
        while (true) {
            try {
                Thread.sleep(delay);
                if (flag == true) {
                    count += 3;
                    repaint();
                    flag = false;
                } else if (deFlag == true) {
                    count -= 5;
                    repaint();
                    deFlag = true;
                } else if (stageFlag == true) {
                    count -= 10;
                    repaint();
                    stageFlag = false;
                } else {
                    count--;
                    if (count <= 0) {
                        new GameOverFrame("timeover",
0, 0);
                        return;
                    }
                }
                text.setText(Integer.toString(count));
            } catch (InterruptedException e) {
                return;
            }
        }
    }
}

```

## 5-10) FeedList.java - 먹이 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class FeedList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("feed.png");
    public FeedList() {
        try {
            Scanner sc = new Scanner(new
FileReader("feedWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getFeedVector() {
        return wordLabel;
    }
}
```

## 5-11) HumList.java - 습도 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class HumList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("hum.png");
    public HumList() {
        try {
            Scanner sc = new Scanner(
new
FileReader("humWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getHumVector() {
        return wordLabel;
    }
}
```

## 5-12) TemperatureList.java - 온도 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class TemperatureList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("tem.png");
    public TemperatureList() {
        try {
            Scanner sc = new Scanner(
                new FileReader("temWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
                JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getTemVector() {
        return wordLabel;
    }
}
```

### 5-13) TimeList.java - 시간 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class TimeList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("time.png");
    public TimeList() {
        try {
            Scanner sc = new Scanner(
                new FileReader("timeWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
                JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getTimeVector() {
        return wordLabel;
    }
}
```



#### 5-14) TreeList.java - 나무 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class TreeList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("tree.png");
    public TreeList() {
        try {
            Scanner sc = new Scanner(
                new FileReader("treeWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
                JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getTreeVector() {
        return wordLabel;
    }
}
```

## 5-15) WaterList.java - 물 단어 리스트 생성

```
package list;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.util.Scanner;
import java.util.Vector;

import javax.swing.ImageIcon;
import javax.swing.JLabel;
import javax.swing.SwingConstants;

public class WaterList {
    private Vector<String> wordVector = new Vector<String>();
    private Vector<JLabel> wordLabel = new Vector<JLabel>();
    ImageIcon image = new ImageIcon("water.png");
    public WaterList() {
        try {
            Scanner sc = new Scanner(
FileReader("waterWords.txt"));
            while(sc.hasNext()) {
                String word = sc.nextLine();
                wordVector.add(word);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
        for(int i=0;i<wordVector.size();i++) {
            JLabel label = new
JLabel(wordVector.get(i),image,SwingConstants.CENTER);
            wordLabel.add(label);
        }
    }
    public Vector<JLabel> getWaterVector() {
        return wordLabel;
    }
}
```

## 6.결론

미니프로젝트를 하면서 자바를 이용하여 파일입출력, 스윙, 그래픽에 대해 배우고 익혔다. 미니프로젝트를 하면서 책을 많이 보고 열심히 찾아보면서 진행하였다. 프로젝트를 하면서 객체 생성과 객체 사용, 스레드에 어려움이 있었고 이러한 문제들을 수업시간에 실습한 프로그램과 책을 이용하여 해결하였다. 그리고 6종류의 아이템을 사용하는 것에 개한 어려움이 제일 컸다. 6개의 스레드를 이용하는 대신 하나의 스레드로 HashMap과 Vector를 이용하여 아이템을 구별해주어 이 문제를 해결하였다. 그리고 객체 생성에 대한 공부의 필요성을 느꼈고 훌륭한 컴퓨터 전문가로서 성장하기 위해 포기하지 않고 계속 공부해가며 오류를 잡아야함을 알게 되었다.