

#길치\_외계인



#우리집찾아줘

#외계인

#외계인도\_집이\_있다

# 외계인 집 찾아주기 게임

2276358 컴퓨터공학과 홍지형



# 목차

---

1.  
외계인  
집 찾아주기 게임  
설명



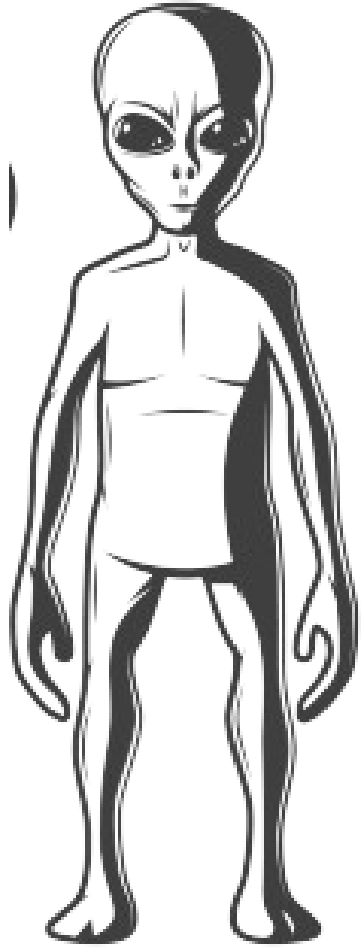
2.  
게임 실행 화면



3.  
소스 코드



# 외계인 집 찾아주기 게임은?



- 길을 잃어버린 길치 외계인을 집에 데려다 주는 설정입니다.
- 방향키**를 입력해서 외계인을 움직일 수 있고  
각 레벨에 해당하는 안내에 따라 외계인을 이동해  
포탈에 접근하여 **엔터키**를 누르면 **레벨을 클리어**할 수 있습니다.



<-외계인

외계인

<-포탈

-이 상태에서  
엔터키를 누르면  
게임을 클리어할 수 있습니다!



# 단계별 게임 클리어 방법



## Level 1

-아무런 장애물이 없습니다.

## Level 2

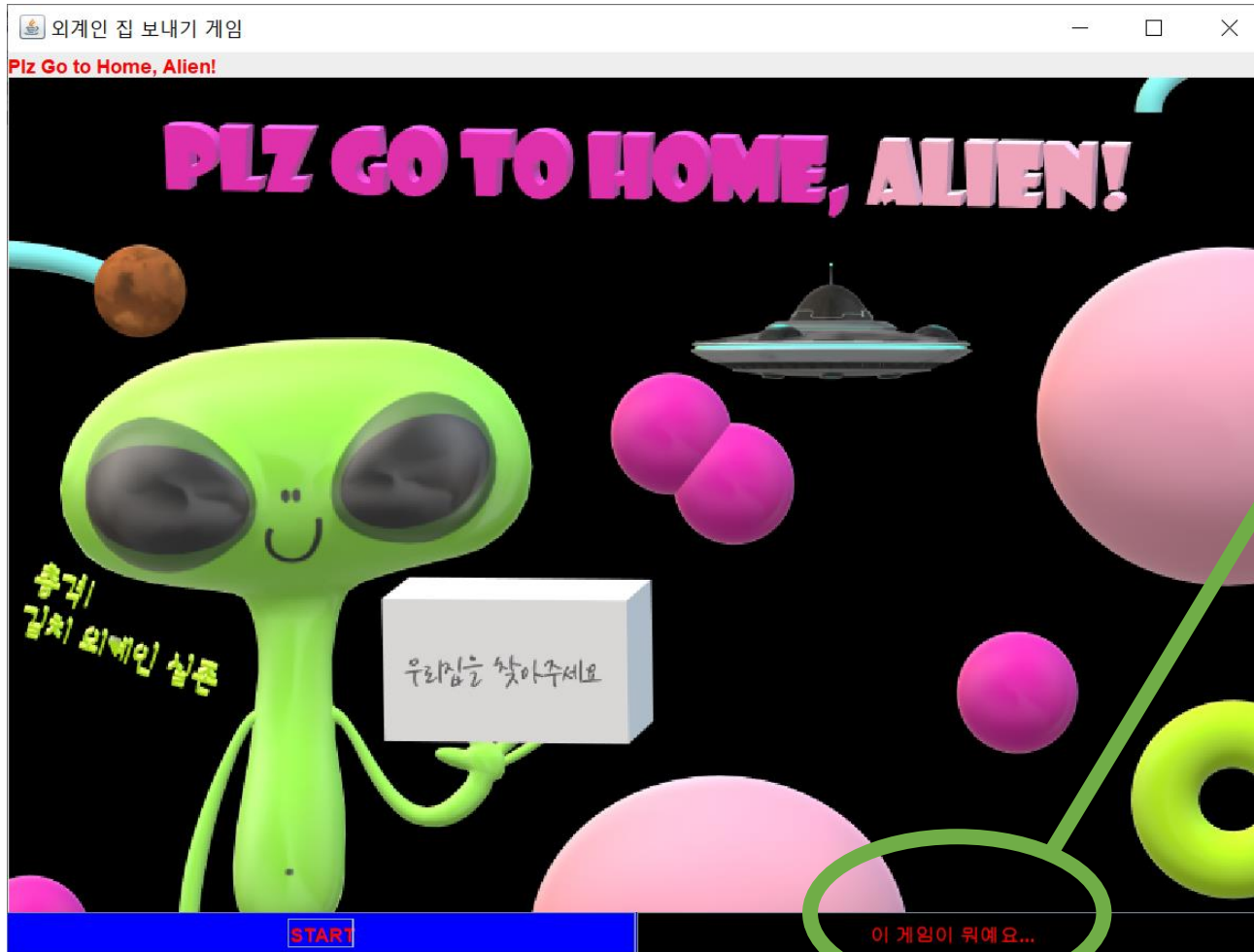
- 보이지 않는 장애물이 있습니다.
- 외계인이 장애물에 닿으면 외계인이 굳습니다.
- 굳은 외계인을 원상 복귀하려면 Reset버튼을 눌러야 합니다.

## Level 3

-포탈이 움직이므로 타이밍을 잘 맞추어 외계인을 움직여야 합니다.

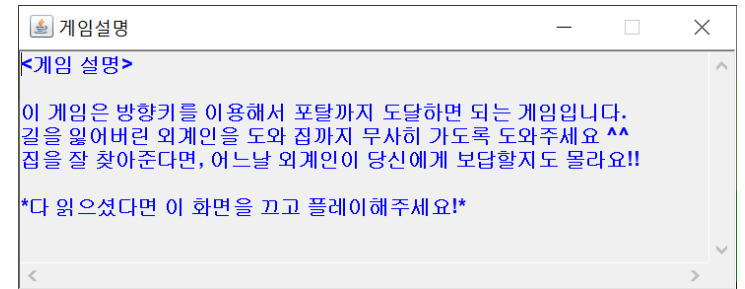


# 게임 실행 화면(1. StartPage)



**'이 게임이 뭐예요...'**

이 버튼을 누르면,  
아래와 같이 게임 방법을  
간략히 소개하는 창이 나옵니다.

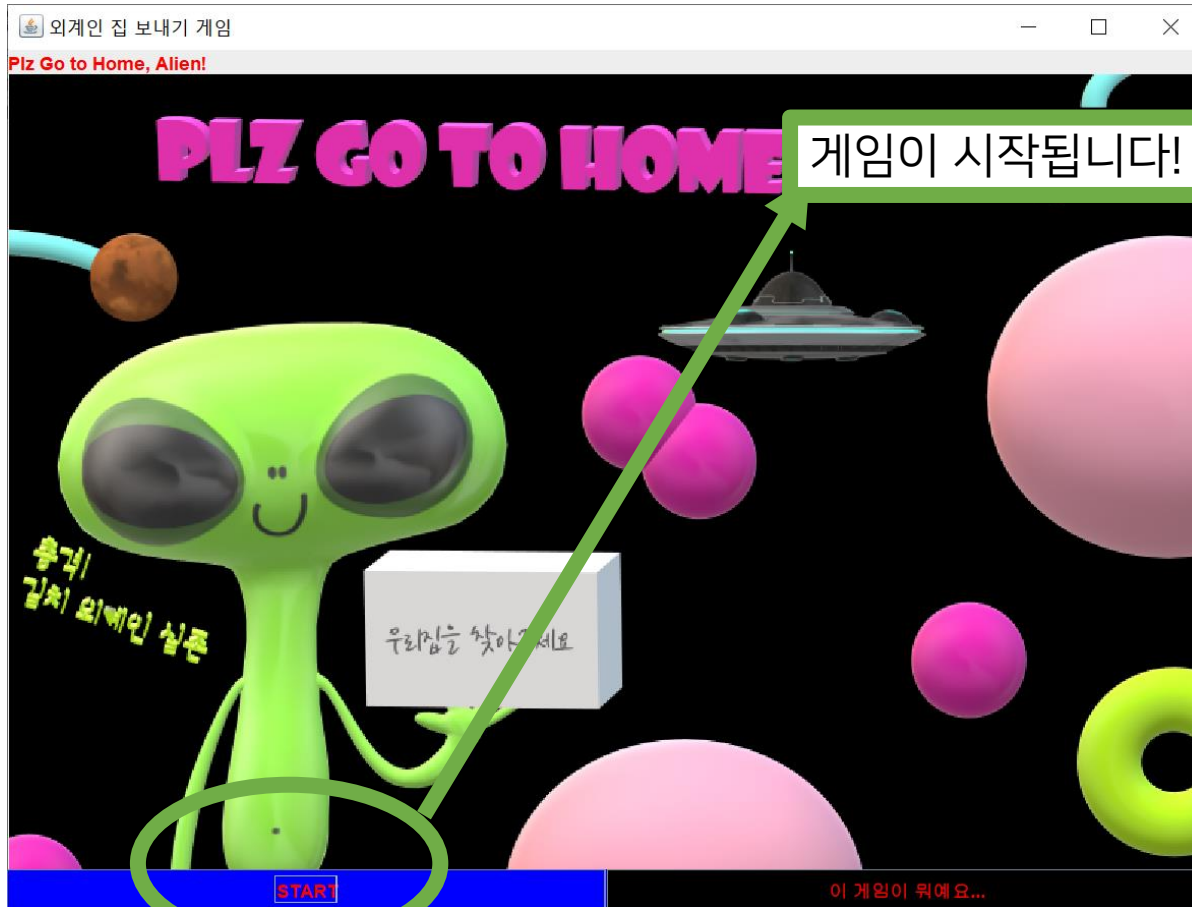


**'게임설명창'**

쉽게 수정이 가능하게끔  
Txt파일을 TextArea에  
출력했습니다.



# 게임 실행 화면



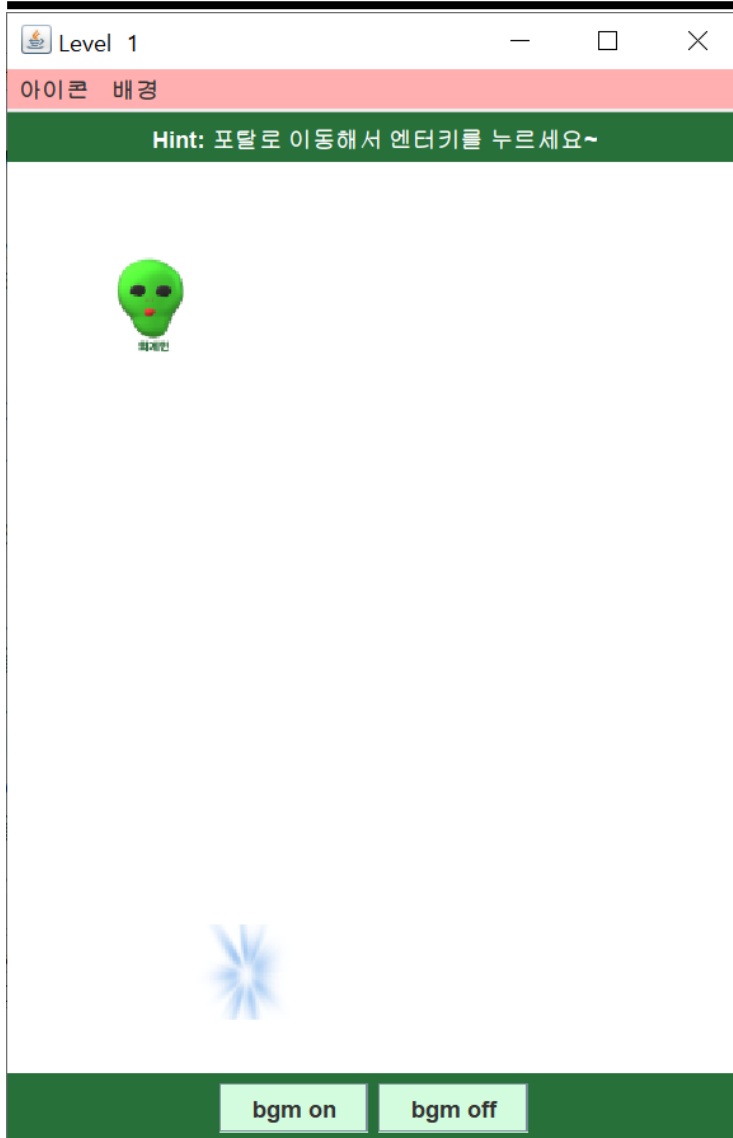
게임이 시작됩니다!

START

이 게임이 뭐예요...



# 게임 실행 화면



## 메뉴바

(아이콘 누르기): 표시되는 외계인의 이미지를 변경할 수 있습니다.  
아래와 같이 총 세 가지의 이미지가 있고  
자유롭게 선택 가능합니다



외계인

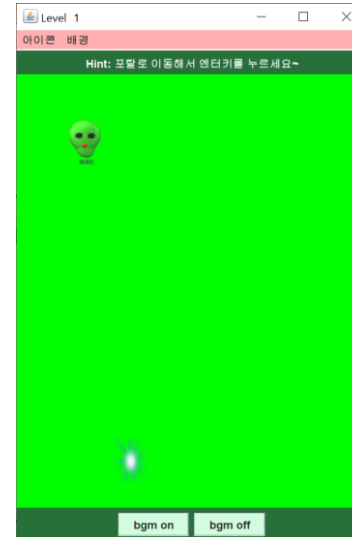
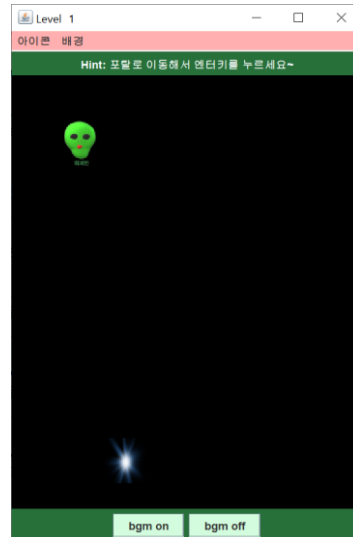


외계인



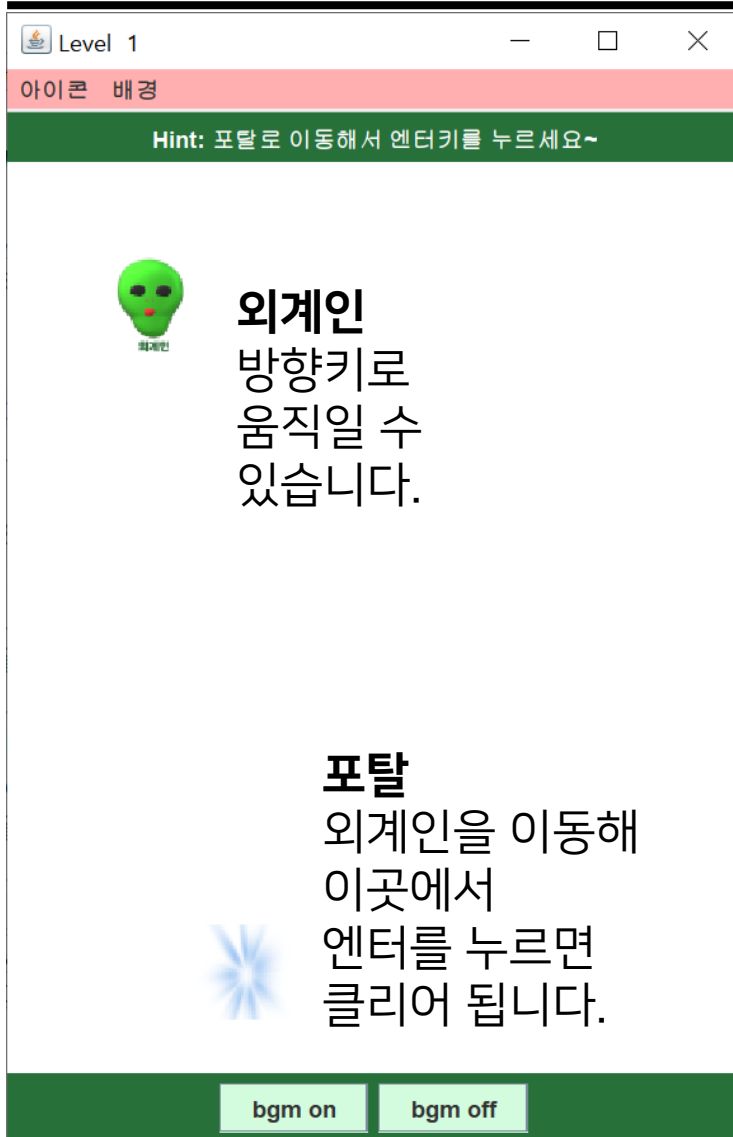
외계인

(배경): 기본, 검정, 녹색을 선택해 배경색을 선택할 수 있습니다.



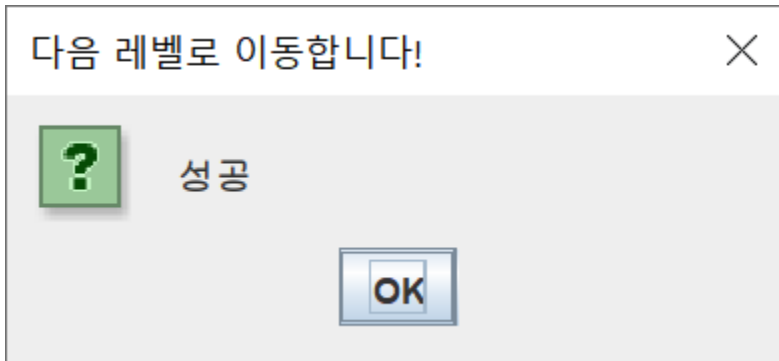


# 게임 실행 화면



## 힌트

어떻게 하면 레벨을 클리어할 수 있는지 알려줍니다.



## 다음 레벨로 이동합니다!

레벨이 클리어 되면 이 창이 뜨고 다음 레벨로 넘어갑니다.

## BGM on/off

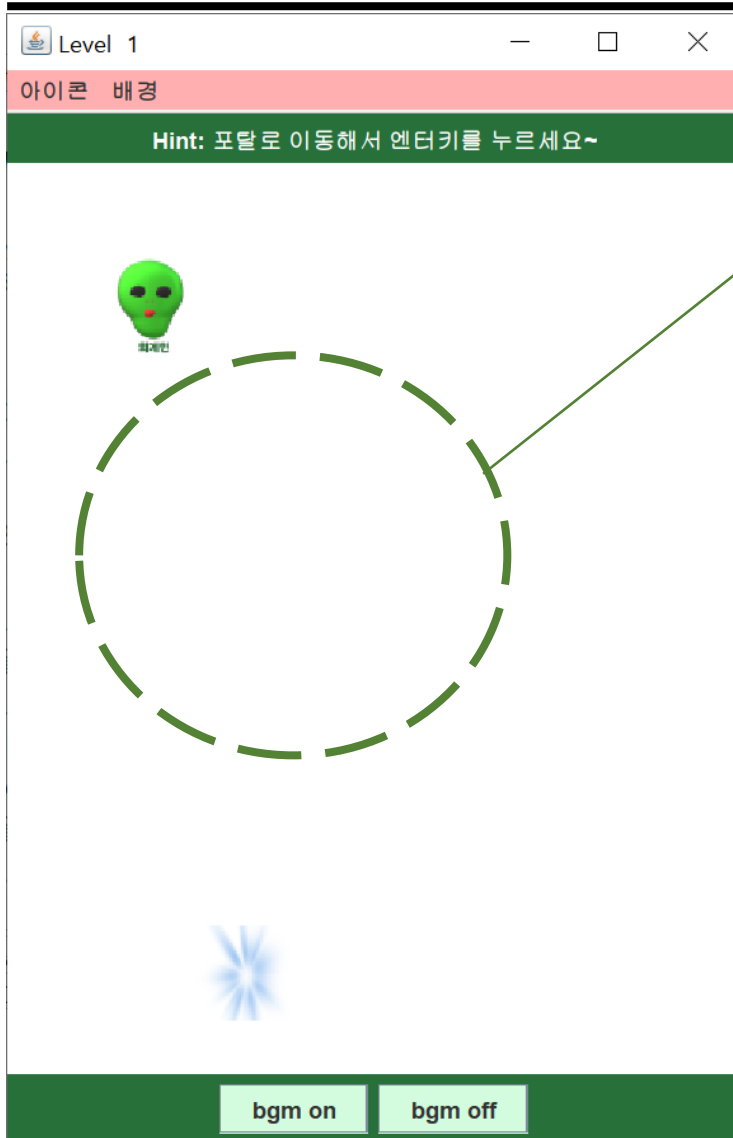
기본적으로 bgm이 나오고 이를 재생/정지할 수 있는 버튼입니다.

Bgm은 나상현씨밴드의 'LoveLoveLove'의 8-bit 버전을 넣어보았습니다.





# 게임 실행 화면



## 장애물

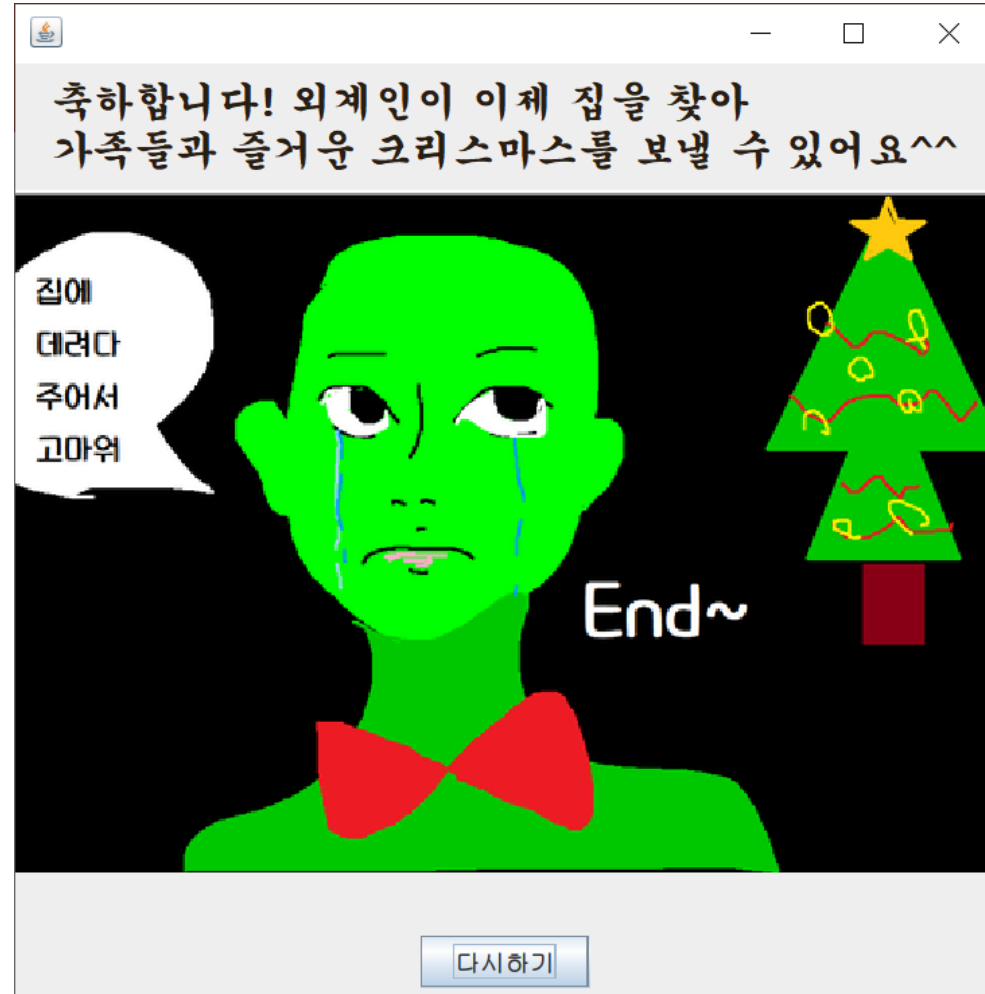
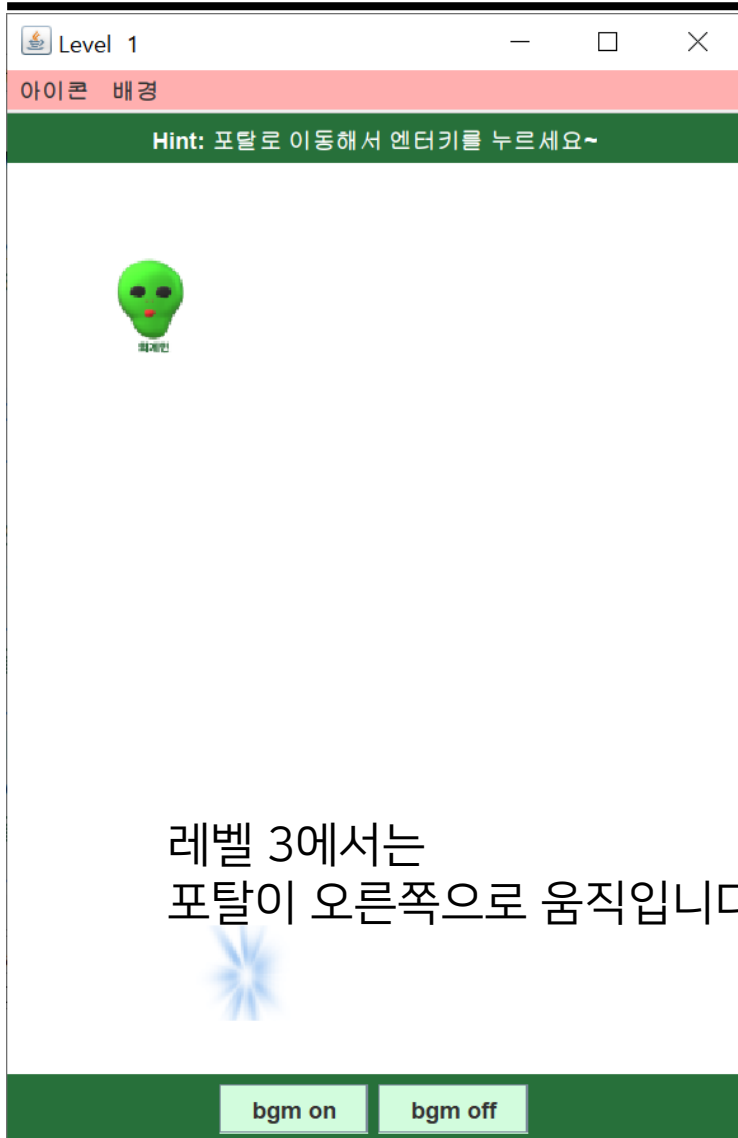
점선으로 표시된 부분에 보이지 않는 장애물이 있어  
피해서 이동해야 합니다.

장애물에 닿으면 외계인이 굳어  
움직이지 않습니다.

이때는 reset 버튼을 눌러 재시작합니다



# 게임 실행 화면



레벨 3을 클리어하면 위와 같은 창이 나오고  
다시하기를 누르면 게임의 첫 화면부터 나오게 됩니다.



# 소스코드(1. StartPage.java)

```
package View;
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;
import java.io.IOException;

public class StartPage extends JFrame{
//처음 시작하는 인트로 부분입니다.

StartPage(){
setTitle("외계인 집 보내기 게임"); //창 제목을 설정
JLabel title = new JLabel("Plz Go to Home, Alien!"); //영어 제목 안내
title.setForeground(Color.red); //영어 제목의 색상 지정
//게임의 첫 화면을 꾸미기 위해서 이미지를 가져왔습니다.
ImageIcon titleImage = new ImageIcon("Image/title.png");
JLabel titleLogo = new JLabel(titleImage);
//버튼을 넣을 패널을 만들고 레이아웃은 그리드로 해주었습니다.
JPanel buttonSet = new JPanel(new GridLayout());
//이 게임을 어떻게 해야하는지 알려주는 버튼을 만들었습니다.
JButton explainBtn = new JButton("이 게임이 뭐예요...");
//버튼을 꾸몄습니다.
explainBtn.setForeground(Color.red);
explainBtn.setBackground(Color.black);
//안내 버튼을 누르면 게임을 안내하는 창인 gameIntro가 나오도록 했습니다.
explainBtn.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
try {
new GameIntro();
} catch (IOException e1) {
// TODO Auto-generated catch block
e1.printStackTrace();
}
```

```
}
}
});
//게임을 시작하는 버튼을 만들고 이 버튼을 누르면 Step1이 나오고
//현재 화면은 없어지도록 만들었습니다.
JButton startBtn = new JButton("START");
startBtn.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e) {
dispose();
new Step1();
}
});
//게임을 시작하는 버튼을 만들었습니다.
startBtn.setBackground(Color.blue);
startBtn.setForeground(Color.red);
//버튼을 넣을 패널에 두개의 버튼을 넣어주었습니다.
buttonSet.add(startBtn);
buttonSet.add(explainBtn);
setSize(1000, 600);
//프레임에 영어제목과 게임이미지, 위에서 만든 버튼 패널을 부착해주었습니다.
add(title, "North");
add(titleLogo, "Center");
add(buttonSet, "South");
setVisible(true);
}

public static void main(String[] args) {
// TODO Auto-generated method stub
new StartPage();
}
}
```



# 소스코드(2. GameIntro.java)

---

```
package View;
import java.awt.*;
import javax.swing.*;

import java.io.*;

public class GameIntro extends JFrame {
    String fullexplain;
    GameIntro() throws IOException, FileNotFoundException {
        FileReader fin = new FileReader("Text/GameExplain.txt");
        TextArea text = new TextArea();
        int ch;
        fullexplain = "";
        System.out.println(fullexplain);
        while((ch=fin.read())!=-1) {
            fullexplain += (char)ch;
        }
        fin.close();
        text.setText(fullexplain);
        Font font = new Font("", Font.BOLD, 13);
        text.setForeground(Color.blue);
        text.setFont(font);
        text.setEditable(false);
        setTitle("게임설명");
        setSize(500, 200);
        add(text, "Center");
        setVisible(true);
        this.setResizable(false);
    }
}
```

```
public static void main(String[] args) throws
IOException{
    // TODO Auto-generated method stub
    GameIntro frame = new GameIntro();

}

}
```



# 소스코드(3. Step1.java)

```
package View;

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

import javax.swing.*;

import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.util.ArrayList;

public class Step1 extends JFrame{
    public JLabel explain;
    public JLabel userLabel;
    public ArrayList<ImageIcon> user;
    public JPanel contentPane;
    public JLabel portallabel;
    public Clip bgmClip;
    public Clip trainEffect;
    public JPanel southPanel;
    public int level;
    boolean BGMstart = false;
```

```
    Step1(){
        setSize(400, 600);
        //창을 띄우면 자동으로 비지엠이 출력되도록 했습니다.
        this.playBGM();
        //레벨을 지정했습니다. 이로써 뒤에 있는 화면 넘어가기에서 각각을 구분을 할 수 있게 됩니다.
        level = 1;
        //자주 쓰이는 컬러를 미리 만들어두었습니다.
        Color deepGreen = new Color(39, 112, 58);
        Color paleGreen = new Color(210, 252, 221);

        //방향키로 움직일 유저 라벨을 만들어줍니다.
        //메뉴에서 유저 라벨의 이미지를 변경할 수 있도록
        //arrayList을 활용해봤습니다.
        user = new ArrayList<>();
        for(int i = 0; i<3; i++) {
            //총 3개의 이미지가 있어 이를 불러왔습니다.
            ImageIcon userImg = new ImageIcon("Image/alienImg"+i+".png");
            Image img = userImg.getImage();
            //이미지 조절이 필요해서 getScaledInstance를 이용해서 조절해주었습니다.
            img = img.getScaledInstance(50, 50, java.awt.Image.SCALE_SMOOTH);
            userImg = new ImageIcon(img);
            //유저 라벨에 이미지를 넣어주었습니다.
            user.add(userImg);
        }
        //borderlayout을 지정했습니다.
        setLayout(new BorderLayout());
        //3개의 panel을 만들어서 구역을 구분해줍니다.
        JPanel northPanel = new JPanel();
        //첫번째 라벨에는 이 레벨에 해당하는 안내를 해주었습니다.
        explain = new JLabel("Hint: 포탈로 이동해서 엔터키를 누르세요~");
        explain.setFont(new Font("", Font.BOLD, 12));
        explain.setForeground(Color.white);
        northPanel.add(explain);
```

```

northPanel.setBackground(deepGreen);
//첫번째 패널을 북쪽에 부착합니다.
add(northPanel, BorderLayout.NORTH);
//유저 라벨이 움직일 가장 중요한 콘텐츠페인을 만들어주었습니다.
contentPane = new JPanel();
contentPane.setBackground(Color.white);
contentPane.setLayout(null);
//이 패널은 가장 센터에 부착했습니다!
add(contentPane, BorderLayout.CENTER);
//방향키로 움직일 유저 라벨을 만들었습니다.
//현재는 가장 기본이 되는 이미지로 라벨을 꾸며놓았습니다.
userLabel = new JLabel(user.get(0));
//유저라벨의 초기 위치를 지정해주고 콘텐츠패널에 놓아주었습니다.
userLabel.setBounds(50, 50, 50, 50);
contentPane.add(userLabel);
//버튼을 놓을 마지막 패널을 생성해줍니다.
southPanel = new JPanel();
southPanel.setBackground(deepGreen);
//센터의 패널에서 유저가 움직여야 하므로
//포커스를 콘텐츠페인에 두었습니다.
contentPane.setFocusable(true);
//방향키로 유저라벨이 움직일 수 있게 키리스터를 부착했습니다
contentPane.addKeyListener(new MoveUser());
//포탈 이미지를 불러와서 라벨에 부착하고 콘텐츠 페인에 놓았습니다.
ImageIcon portalImg = new ImageIcon("Image/portalImg.png");
Image porImg = portalImg.getImage();
porImg = porImg.getScaledInstance(50, 50, java.awt.Image.SCALE_SMOOTH);
portalImg = new ImageIcon(porImg);
portalLabel = new JLabel(portalImg);
portalLabel.setBounds(100, 400, 50, 50);
contentPane.add(portalLabel);
//비지엄을 재생하는 버튼을 만들었습니다.
JButton bgmOnBtn = new JButton("bgm on");
bgmOnBtn.setBackground(paleGreen);
//비지엄을 끄는 버튼을 만들었습니다.
JButton bgmOffBtn = new JButton("bgm off");

```

```

//비교적 간단한 부분이라 익명클래스를 이용해 키리스너를 부착했습니다.
bgmOnBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        //비지엄 재생버튼을 누르면 유저라벨이 안 움직여서
        //포커스를 조절해주었습니다.
        bgmOnBtn.setFocusable(false);
        bgmOffBtn.setFocusable(false);
        //다시 음악을 켜줍니다.
        if(!BGMstart)
            playBGM();
    }
});

//마찬가지로 익명클래스를 이용해서 부착합니다.
bgmOffBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        bgmOnBtn.setFocusable(false);
        bgmOffBtn.setFocusable(false);
        //음악을 끄는 함수를 넣었습니다.

        stopBGM();
    }
});
//버튼 색을 꾸밈습니다.
bgmOffBtn.setBackground(paleGreen);
//가장 아래 패널에 버튼 2개를 넣었습니다.
southPanel.add(bgmOnBtn);
southPanel.add(bgmOffBtn);
add(southPanel, BorderLayout.SOUTH);

setTitle("Level 1");
//메뉴를 부착합니다.
createMenu();
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

setVisible(true);
}

```

```

//가장 윗부분에 메뉴를 보여주는 함수입니다.
public void createMenu() {
    JMenuBar menuBar = new JMenuBar();
    MenuAction act = new MenuAction();
    menuBar.setBackground(Color.pink);
    //메뉴창에서 아이콘을 누르면 유저 라벨의 이미지를 변경할 수 있게 했습니다.
    JMenu iconMenu = new JMenu("아이콘");
    JMenuItem[] userItems = new JMenuItem[user.size()];
    for(int i = 0; i<user.size(); i++) {
        userItems[i] = new JMenuItem("테마"+i);
        userItems[i].addActionListener(act);
        iconMenu.add(userItems[i]);
        //구분선을 넣어주었습니다.
        iconMenu.addSeparator();
    }
    //배경색을 바꿀 수 있는 메뉴를 생성했습니다.
    JMenu backMenu = new JMenu("배경");

    String[] items = {"기본", "검정", "녹색"};
    JMenuItem[] bmenuItems = new JMenuItem[3];
    for(int i = 0; i<items.length; i++) {
        bmenuItems[i] = new JMenuItem(items[i]);
        bmenuItems[i].addActionListener(act);
        backMenu.add(bmenuItems[i]);
        backMenu.addSeparator();
    }
    //메뉴바에 두개의 메뉴를 부착했습니다.
    menuBar.add(iconMenu);
    menuBar.add(backMenu);
    setJMenuBar(menuBar);
}

```

//메뉴를 누르면 각각의 기능을 구현할 수 있게 해줍니다.

```

class MenuAction implements ActionListener{
    public void actionPerformed(ActionEvent e) {

        String command = e.getActionCommand();

        switch(command) {
            //배경에서 기본을 누르면 배경색이 하얀색이 됩니다.
            case "기본":
                if(contentPane.getBackground() != Color.white)
                    contentPane.setBackground(Color.white);
                break;
            //배경에서 검정을 누르면 배경색이 검은색이 됩니다.
            case "검정":
                if(contentPane.getBackground() != Color.black)
                    contentPane.setBackground(Color.black);
                break;
            //배경에서 녹색을 누르면 배경색이 녹색이 됩니다.
            case "녹색":
                if(contentPane.getBackground() != Color.green)
                    contentPane.setBackground(Color.green);
                break;
            //유저라벨을 다른 이미지로 변경할 수 있게 했습니다!
            case "테마0":
                userLabel.setIcon(user.get(0));
                break;
            case "테마1":
                userLabel.setIcon(user.get(1));
                break;
            case "테마2":
                userLabel.setIcon(user.get(2));
                break;
        }
    }
}

```

```

//움직일 수 있는 위치인지 판단해주는 함수입니다.
//여기선 장애물이 없어 true로 리턴해줍니다.

boolean canMove(int x, int y) {
return true;
}

//유저라벨을 방향키로 움직일 수 있게 하는 내부클래스입니다.
class MoveUser extends KeyAdapter{
public void keyPressed(KeyEvent e) {
//현재 유저라벨의 위치를 알아냅니다.
int x = userLabel.getX();
int y = userLabel.getY();
//방향키를 누를 때마다 변경할 값을 미리 지정합니다.
int moveNum = 5;
//각각의 방향키를 누를때마다 유저라벨의 위치를 변경해줍니다.
if(canMove(0, 0)) {
switch(e.getKeyCode()) {
case KeyEvent.VK_UP:
userLabel.setLocation(x, Math.max(y - moveNum, 0));
break;
case KeyEvent.VK_DOWN:
userLabel.setLocation(x, Math.min(y + moveNum, getHeight() -
userLabel.getHeight()));
break;
case KeyEvent.VK_LEFT:
userLabel.setLocation(Math.max(x - moveNum, 0), y);
break;
case KeyEvent.VK_RIGHT:
userLabel.setLocation(Math.min(x + moveNum, getWidth() -
userLabel.getWidth()), y);
break;
//포탈 위에서 엔터를 누르면 다음 레벨로 넘어갑니다.
case KeyEvent.VK_ENTER:
if( level ==1 &&isLabelOverlapping(userLabel, portallLabel) ) {
JOptionPane.showConfirmDialog(null, "성공", "다음 레벨로 이동합니다!",
JOptionPane.PLAIN_MESSAGE);
new Step2();
dispose();
}
}
}
}

//포탈에 유저 라벨이 올라가있는지 판단하기 위한 함수를 만들었습니다.
boolean isLabelOverlapping(JLabel label1, JLabel label2) {
int x1 = label1.getX();
int y1 = label1.getY();
int width1 = label1.getWidth();
int height1 = label1.getHeight();

int x2 = label2.getX();
int y2 = label2.getY();
int width2 = label2.getWidth();
int height2 = label2.getHeight();

return x1 < x2 + width2 && x1 + width1 > x2 && y1 < y2 + height2 && y1 +
height1 > y2;
}

//오디오스트림을 활용해서 bgm을 트는 함수입니다.
public void playBGM() {
try {
// 파일을 불러와줍니다.
AudioInputStream audioStream =
AudioSystem.getAudioInputStream(getClass().getResource("/Music/Love3.wav"));

// 오디오 재생을 위한 클립을 만듭니다.
bgmClip = AudioSystem.getClip();
bgmClip.open(audioStream);

// 무한으로 재생되게 합니다.
bgmClip.loop(Clip.LOOP_CONTINUOUSLY);

// 이제 진짜로 재생되게 합니다.
BGMstart = true;
bgmClip.start();
} catch (UnsupportedAudioFormatException | IOException | LineUnavailableException
e) {
e.printStackTrace();
}
}
}

```



//비지엠을 종료하는 함수입니다.

```
public void stopBGM() {  
    if (bgmClip != null && bgmClip.isRunning())  
{  
        BGMstart = false;  
        bgmClip.stop();  
        bgmClip.close();  
    }  
}
```

//창을 종료할때 기존의 비지엠도 종료해야하므로 이렇게 오버라이딩 시켜주었습니다.

```
public void dispose() {  
    stopBGM();  
    super.dispose();  
}
```

```
public static void main(String[] args) {  
    // TODO Auto-generated method stub  
    new Step1();  
}  
  
}
```



# 소스코드(4. Step2.java)

```
package View;

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

import javax.swing.*;

import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.util.ArrayList;

public class Step2 extends Step1 {
    //레벨2에서는 장애물을 생성해서 난이도를 높였습니다.
    ArrayList<JLabel> bars;

    Step2(){
        //기존의 step1을 재활용하기 위해 상속받고 생성자도 그대로 불러왔습니다.
        super();
        //레벨2로 넘어갔다는 표시를 하기 위해서 변경했습니다.
        level =2;
        setTitle("Level 2");
        //장애물을 어레이리스트로 만들어주었습니다.
        bars = new ArrayList<>();
        for(int i = 0; i <3; i++) {
            JLabel bar = new JLabel();

            bars.add(bar);
        }

        //장애물의 크기와 위치를 설정하고 패널에 부착합니다.
        for(int i= 0; i<bars.size(); i++) {
            bars.get(i).setBounds(80+i*20, 130+i*50, 100, 30);
            contentPane.add(bars.get(i));
        }
        //장애물에 닿으면 사용자가 움직이는 않는 설정이기 때문에 리셋버튼이 필요했습니다
        //리셋버튼을 누르면 다시 step2가 켜지도록 했습니다.
        JButton resetBtn = new JButton("reset");
        resetBtn.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                dispose();
                new Step2();
            }
        });
        //reset버튼을 기존 버튼패널에 부착했습니다.
        southPanel.add(resetBtn);
        //레벨2에 해당하는 안내를 해줍니다.
        explain.setText("<html> Hint : 보이지 않는 장애물을 피해 포탈로 이동하세요!<br> 장애물에  
닿으면 외계인은 굳어버려요!</html>");
        //이전과는 다른 부분이 있기에 새 키리스너를 부착했습니다.
        contentPane.addKeyListener(new Moveuser2());

    }
    //장애물과 닿으면 false, 아니면 true를 리턴하는 함수입니다.
    boolean canMove(int x, int y) {
        for (JLabel bar : bars) {
            if (isLabelOverlapping(userLabel, bar)) {
                return false;
            }
        }
        return true;
    }
}
```

//여기서는 기존의 무브유저를 이용하되 새부분을 추가할 수 있게 상속 받았습니다.

```
class Moveuser2 extends MoveUser{  
public void keyPressed(KeyEvent e) {
```

```
int x= userLabel.getX();
```

```
int y= userLabel.getY();
```

//변경된 부분인데, 엔터를 누르면 현재창을 종료하고 레벨3으로 넘어갑니다.

```
if(e.getKeyCode() == KeyEvent.VK_ENTER) {
```

```
if( isLabelOverlapping(userLabel, portalLabel) )
```

```
{
```

```
dispose();
```

```
new Step3();
```

```
}
```

```
}
```

//장애물이 없을 때만 이동할 수 있게 했습니다.

```
if(canMove(x, y))
```

```
super.keyPressed(e);
```

```
}
```

```
}
```

```
public static void main(String[] args) {
```

```
// TODO Auto-generated method stub
```

```
new Step2();
```

```
}
```

```
}
```



# 소스코드(5. Step3.java)

```
package View;

import javax.sound.sampled.AudioInputStream;
import javax.sound.sampled.AudioSystem;
import javax.sound.sampled.Clip;
import javax.sound.sampled.LineUnavailableException;
import javax.sound.sampled.UnsupportedAudioFileException;

import javax.swing.*;

import View.Step1.MoveUser;
import View.Step2.Moveuser2;

import java.awt.*;
import java.awt.event.*;
import java.io.IOException;
import java.util.ArrayList;

public class Step3 extends Step1{
//타이머를 이용해서 포탈이 움직이게 해줍니다.
Timer portalTimer;

Step3(){
//스텝1을 재할용합니다.
super();
level = 3;
//레벨3에 맞는 타이틀, 안내문을 변경합니다.
setTitle("Level 3");
explain.setText("Hint: 이제는 움직이는 포탈로 들어가야 해요~");
```

```
//포탈이 움직일 수 있게 액션리스너를 부착합니다.
portalTimer = new Timer(100, new ActionListener() {
public void actionPerformed(ActionEvent e) {
movePortal();
}
});
portalTimer.start();
//기존과 달라진 부분이 있어서 새 키리스너 부착
contentPane.addKeyListener(new MoveUser3());
}
//포탈을 움직이는 함수입니다.
public void movePortal() {
int speed = 7;
int x = portalLabel.getX();
int y = portalLabel.getY();
//오른쪽으로 움직이도록 했습니다.
x += speed;
//화면에서 포탈이 벗어나면 안되니까 포탈이 화면 내에 있도록 지정했습니다.
if(x >getWidth()) {
x = -portalLabel.getWidth();
}
portalLabel.setLocation(x, y);
}
```

```
class MoveUser3 extends MoveUser{
public void keyPressed(KeyEvent e) {
int x= userLabel.getX();
int y= userLabel.getY();
//포털에서 엔터를 누르면 최종 창으로 넘어가게 했습니다.
if(e.getKeyCode() == KeyEvent.VK_ENTER) {
if( isLabelOverlapping(userLabel, portalLabel) )
{
dispose();
new LastPage();
}
}
super.keyPressed(e);
}
}

public static void main(String [] args) {
new Step3();
}
}
```



# 소스코드(5. LastPage.java)

---

```
package View;

import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

public class LastPage extends JFrame{

    LastPage(){
        setSize(500, 500);
        this.setLayout(new BorderLayout());
        //종료를 안내하는 문구를 나타내는 부분을 만들었습니다.
        JPanel firstPanel = new JPanel();
        add(firstPanel, "North");
        JLabel mess = new JLabel("<html>축하합니다! 외계인이 이제 집을 찾아<br>가족들과 즐거운 크리스마스를 보낼 수 있어요^^</html>");
        mess.setForeground(new Color(39, 29, 19));
        mess.setFont(new Font("궁서", Font.BOLD, 20));
        firstPanel.add(mess);
        //기쁨의 눈물을 흘리는 외계인 그림을 넣어주었습니다.
        JPanel secondPanel = new JPanel();
        ImageIcon lastImg = new ImageIcon("Image/bye.png");
```

```
JLabel byeLabel = new JLabel(lastImg);
secondPanel.add(byeLabel);
//버튼을 부착할 수 있게 패널을 만들었습니다.
JPanel lastPanel = new JPanel();
JButton reBtn = new JButton("다시하기");
//버튼을 누르면 다시 게임을 시작할 수 있습니다.
reBtn.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        dispose();
        new StartPage();
    }

});
lastPanel.add(reBtn);

add(secondPanel, "Center");
add(lastPanel, "South");
setVisible(true);
}

public static void main(String[] args) {
    // TODO Auto-generated method stub
    new LastPage();
}

}
```

감사합니다😊  
즐거운 크리스마스 보내세요!!

