

คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของวิชา Programming Fundamentals II มีเนื้อเกี่ยวกับรายละเอียดของ
โครงการเกี่ยวกับรายวิชาวิชาที่ได้เรียน ผู้จัดทำได้ทำเกมเพื่อทดสอบทักษะภาษาอังกฤษเกี่ยวกับสำนวน ที่ผู้จัดทำ
สนใจ คือ เกม Idioms Quiz

ผู้จัดทำคาดหวังว่ารายงานฉบับนี้จะเป็นประโยชน์กับผู้อ่าน หากมีข้อผิดพลาดประการใด ผู้จัดทำขอน้อม
รับและขออภัยมา ณ ที่นี้ด้วย

ผู้จัดทำ

สารบัญ

เรื่อง	หน้า
คุณสมบัติของเกม	1
หลักการออกแบบเกม	1
ตัวอย่างการรันโปรแกรม	2
Source Code	4

Idioms Quiz

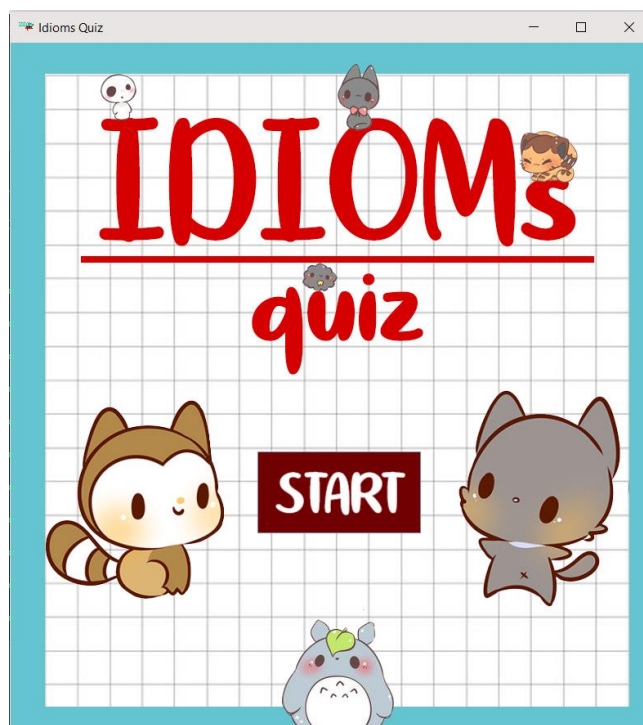
คุณสมบัติของเกม

เกมนี้เป็นเกมที่สร้างขึ้นเพื่อทดสอบทักษะภาษาอังกฤษ ที่เกี่ยวกับสำนวน ซึ่งตัวเกมจะประกอบไปด้วย ส่วนคำถาม และส่วนคำตอบอีก 3 อัน ผู้เล่นมีโอกาสในการตอบผิดได้เพียง 2 ครั้ง เมื่อถึงครั้งที่ 3 เกมจะจบ หลังจากนั้นผู้เล่นสามารถเลือกที่จะเล่นอีกครั้ง หรือจะออกเกมเลยก็ได้ ตัวคำถามและคำตอบจะเป็นสุ่มขึ้น คำตอบ จะมีรูปคำตอบที่ถูกอยู่ 3 ที่ซึ่งจะเป็นการสุ่มเช่นกัน คำถามของเกมนี้จะมีเพียง 49 คำถามเมื่อผู้เล่น สามารถเล่นได้ครบและหัวใจยังไม่หมดจะขึ้นหน้า win game

หลักการออกแบบเกม

- **การสุ่ม** จะมีสุ่ม 2 อย่าง คือ การสุ่มรูปแบบของคำตอบ และ การสุ่มตำแหน่ง เพื่อนำตำแหน่งมาใช้ในการสร้างคำถาม และ คำตอบ
- **การใส่คำถามและคำตอบ** ในที่นี้ผู้จัดทำ ได้ใช้อาร์เรย์ลิสต์ ในการใส่รูปของคำถามและคำตอบซึ่งมีทั้งหมด 4 อาร์เรย์ลิสต์ ประกอบด้วย คำถาม คำตอบที่ถูก และ คำตอบที่ผิดอีก 2 อาร์เรย์ลิสต์
- **หัวใจ** เมื่อเริ่มเกมผู้เล่นจะมีหัวใจ 3 ดวง และทุกการกดจะมีการเช็คความผิดหรือไม่ ถ้ากดผิดหัวใจจะลดลง แต่ถ้าถูกหัวใจจะเท่าเดิม
- **การกด** เมื่อผู้เล่นกดปุ่มใดๆ อีเว้นท์ในปุ่มแต่ละปุ่มจะมี ดังนี้
 - เช็คค่าผู้เล่นกดถูกหรือไม่ ถ้าถูกแล้วจะทำการสุ่มรูปแบบ และ ตำแหน่งของข้อต่อไปเลย แต่ถ้าผิดหัวใจของผู้เล่นจะลดลง
 - เมื่อสุ่มตำแหน่งของข้อถัดไปเสร็จแล้ว จะทำการใส่รูปใน Label ที่เป็นส่วนของคำถาม และ Button ทั้ง 3 อันที่เป็นส่วนคำตอบ

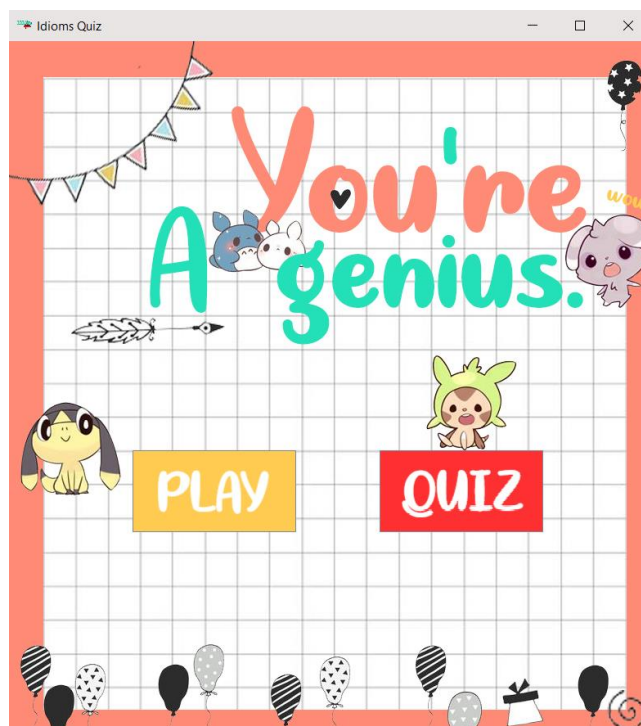
ตัวอย่างการรันโปรแกรม



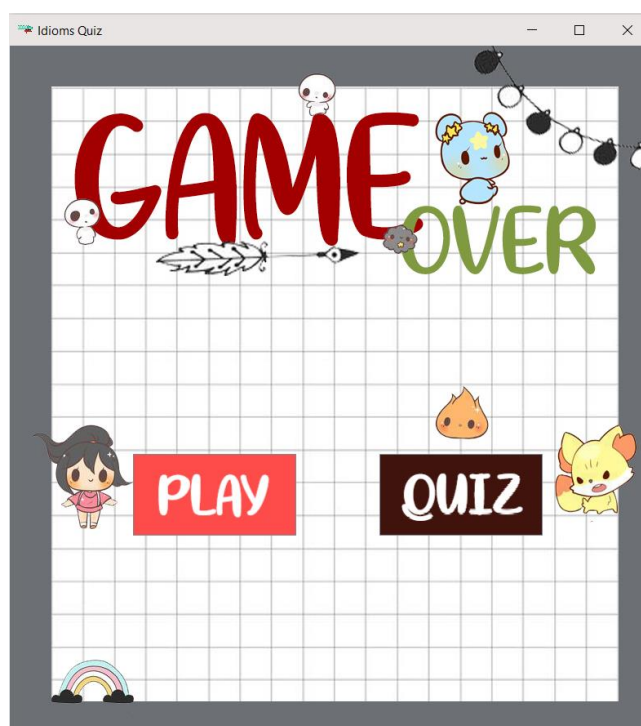
หน้าเริ่มเกม



หน้าเกม



หน้านี้จะแสดงเมื่อผู้เล่นตอบครบทุกคำถาม



หน้านี้จะแสดงเมื่อผู้เล่นตอบผิดจนหัวใจหมด

Source Code

File : Game.java

```
public class Game{  
    public static void main(String[] args) {  
        new GUIGame();  
    }  
}
```

File : GUIGame.java

```
import java.util.Random;
import java.util.ArrayList;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.URL;
import javax.sound.sampled.*;

public class GUIGame{
    private JFrame f = new JFrame("Idioms Quiz");
    private int heart = 2;
    private int choice;
    private int pos;
    private int choose;
    private ImageIcon imgicon = new ImageIcon("Resources/Logo/logo2.png");
    private ArrayList<String> questionList = new ArrayList<String>();
    private ArrayList<String> answer1List = new ArrayList<String>();
    private ArrayList<String> answer2List = new ArrayList<String>();
    private ArrayList<String> answer3List = new ArrayList<String>();
    private ArrayList<Integer> use = new ArrayList<Integer>();
    private ArrayList<Integer> used = new ArrayList<Integer>();
    private JLabel backGround = new JLabel();
    private JLabel Heartt = new JLabel();
    private JLabel Question = new JLabel ();
    private JButton Answer1 = new JButton ();
    private JButton Answer2 = new JButton ();
```



```

private JButton Answer3 = new JButton ();

private Clip onGame;

private Clip right;

private Clip wrong;

public GUIGame() {

    f.setIconImage(imgicon.getImage());
    f.setSize(805, 890);
    f.setLocation(500, 50);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.setLayout( null );
    playMusicGame();

    //set Background
    backGround.setLocation(0,0);
    backGround.setSize(850,850);
    backGround.setIcon(new ImageIcon ("Resources/Background/bg11.jpg"));

    Heartt.setLocation(320,20);
    Heartt.setSize(200,50);
    Heartt.setBackground(Color.BLACK);

    //create button question
    Color c0 = new Color(108,1,2);
    Question.setLocation ( 90, 100 );
    Question.setSize (600,100);
    Question.setBackground (c0);
    Question.setOpaque(true);

```

```
//create button answer1
Color c1 = new Color(166,180,1);
Answer1.setLocation ( 90, 300 );
Answer1.setSize (600,100);
Answer1.setBackground (c1);

//create button answer2
Color c2 = new Color(190,1,2);
Answer2.setLocation ( 90, 450 );
Answer2.setSize (600,100);
Answer2.setBackground (c2);

//create button answer3
Color c3 = new Color(239,246,123);
Answer3.setLocation ( 90, 600 );
Answer3.setSize (600,100);
Answer3.setBackground (c3);

uselist();
fillQuestionList();
fillAnswerList1();
fillAnswerList2();
fillAnswerList3();

//startMenu();
f.add(Heartt);
f.add(Question);
```

```

f.add(Answer1);
f.add(Answer2);
f.add(Answer3);
f.add(backGround);

drawHeart(heart);
choice = ranPattern();
pos = ranPos(use);
drawQuestion(pos,questionList);
drawAns(choice,pos,answer1List,answer2List,answer3List);

f.setVisible(true);

//action button answer 1
Answer1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        //answer is Right
        choose = 1;
        if(choose == choice){
            playMusicOnClickRight();
            drawHeart(heart);
            removerQues(pos,questionList);
            removerAns1(pos,answer1List);
            removerAns2(pos,answer2List);
            removerAns3(pos,answer3List);
            choice = ranPattern();
            pos = ranPos(use);
            drawQuestion(pos,questionList);

```

```

        drawAns(choice,pos,answer1List,answer2List,answer3List);
    }
    else{
        playMusicOnClickWrong();
        if(heart != 0){
            heart -= 1;
            drawHeart(heart);
        }
        else{
            drawHeart(heart);
            if (onGame.isRunning()) onGame.stop();
            new overState();
            f.dispose();
        }
        removerQues(pos,questionList);
        removerAns1(pos,answer1List);
        removerAns2(pos,answer2List);
        removerAns3(pos,answer3List);
        choice = ranPattern();
        pos = ranPos(use);
        drawQuestion(pos,questionList);
        drawAns(choice,pos,answer1List,answer2List,answer3List);
    }
}
});

//action button answer 2
Answer2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {

```

```
//answer is Wrong
choose = 2;
if(choose == choice){
    playMusicOnClickRight();
    drawHeart(heart);
    removerQues(pos,questionList);
    removerAns1(pos,answer1List);
    removerAns2(pos,answer2List);
    removerAns3(pos,answer3List);
    choice = ranPattern();
    pos = ranPos(use);
    drawQuestion(pos,questionList);
    drawAns(choice,pos,answer1List,answer2List,answer3List);
}
else{
    playMusicOnClickWrong();
    if(heart != 0){
        heart -= 1;
        drawHeart(heart);
    }
    else{
        drawHeart(heart);
        if (onGame.isRunning()) onGame.stop();
        new overState();
        f.dispose();
    }
    removerQues(pos,questionList);
    removerAns1(pos,answer1List);
```

```

        removerAns2(pos,answer2List);
        removerAns3(pos,answer3List);
        choice = ranPattern();
        pos = ranPos(use);
        drawQuestion(pos,questionList);
        drawAns(choice,pos,answer1List,answer2List,answer3List);
    }
}
});

```

```

//action button answer 3

```

```

Answer3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        choose = 3;
        if(choose == choice){
            playMusicOnClickRight();
            drawHeart(heart);
            removerQues(pos,questionList);
            removerAns1(pos,answer1List);
            removerAns2(pos,answer2List);
            removerAns3(pos,answer3List);
            choice = ranPattern();
            pos = ranPos(use);
            drawQuestion(pos,questionList);
            drawAns(choice,pos,answer1List,answer2List,answer3List);
        }
        else{
            playMusicOnClickWrong();

```

```

    if(heart != 0){
        heart -= 1;
        drawHeart(heart);
    }
    else{
        drawHeart(heart);
        if (onGame.isRunning()) onGame.stop();
        new overState();
        f.dispose();
    }
    removerQues(pos,questionList);
    removerAns1(pos,answer1List);
    removerAns2(pos,answer2List);
    removerAns3(pos,answer3List);
    choice = ranPattern();
    pos = ranPos(use);
    drawQuestion(pos,questionList);
    drawAns(choice,pos,answer1List,answer2List,answer3List);
}
}
});
}

```

```

public void drawHeart(int heart){
    if(heart == 2){
        Heartt.setIcon(new ImageIcon ("Resources/Heart/h1.png"));
    }
    else if(heart == 1){

```

```

        Heartt.setIcon(new ImageIcon ("Resources/Heart/h2.png"));
    }
    else if(heart == 0){
        Heartt.setIcon(new ImageIcon ("Resources/Heart/h3.png"));
    }
}

//method random pattern of answer
public int ranPattern(){
    Random rand = new Random();
    return choice = rand.nextInt(100)%3+1;
}

//method random position of Question,Answer1,Answer2,Answer3
public int ranPos(ArrayList<Integer> use){
    Random rand = new Random();
    if(use.size() != 0){
        pos = rand.nextInt(100)%(use.size());
        use.remove(pos);
    }
    else{
        if (onGame.isRunning()) onGame.stop();
        new winState();
        f.dispose();
    }
    return pos;
}

```



```
//add value of use 1-50
```

```
public ArrayList<Integer> uselist(){
    for (int i=0;i <= 3 ;i++ ) {
        use.add(i);
    }
    return(use);
}
```

```
//add image of Question in questionList and getImage of questionList
```

```
public ArrayList<String> fillQuestionList(){
    for (int i=1;i<=5;i++ ) {
        questionList.add("Resources/Question/q" +String.valueOf(i)+ ".png");
    }
    return(questionList);
}
```

```
public void drawQuestion(int pos,ArrayList<String> questionList){
    Question.setIcon ( new ImageIcon (questionList.get(pos)));
}
```

```
// add image of Answer1 in answer1List
```

```
public ArrayList<String> fillAnswerList1(){
    for (int i=1;i<=5;i++ ) {
        answer1List.add("Resources/Answer/an1" +String.valueOf(i)+ ".png");
    }
    return(answer1List);
}
```

```
// add image of Answer2 in answer2List
public ArrayList<String> fillAnswerList2(){
    for (int i=1;i<=5;i++ ) {
        answer2List.add("Resources/Answer/an2" +String.valueOf(i)+ ".png");
    }
    return(answer2List);
}
```

```
// add image of Answer3 in answer3List
public ArrayList<String> fillAnswerList3(){
    for (int i=1;i<=5;i++ ) {
        answer3List.add("Resources/Answer/an3" +String.valueOf(i)+ ".png");
    }
    return(answer3List);
}
```

```
public void drawAns(int choice,int pos,ArrayList<String> answer1List,ArrayList<String>
answer2List,ArrayList<String> answer3List){
    if(choice == 1){
        Answer1.setIcon ( new ImageIcon (answer1List.get(pos)));
        Answer2.setIcon ( new ImageIcon (answer2List.get(pos)));
        Answer3.setIcon ( new ImageIcon (answer3List.get(pos)));
    }
    else if(choice == 2){
        Answer2.setIcon ( new ImageIcon (answer1List.get(pos)));
        Answer1.setIcon ( new ImageIcon (answer2List.get(pos)));
        Answer3.setIcon ( new ImageIcon (answer3List.get(pos)));
    }
    else if(choice == 3){
```

```

        Answer3.setIcon ( new ImageIcon (answer1List.get(pos)));
        Answer1.setIcon ( new ImageIcon (answer2List.get(pos)));
        Answer2.setIcon ( new ImageIcon (answer3List.get(pos)));
    }
}

public ArrayList<String> removerQues(int pos,ArrayList<String> questionList){
    questionList.remove(pos);
    return (questionList);
}

public ArrayList<String> removerAns1(int pos,ArrayList<String> answer1List){
    answer1List.remove(pos);
    return (answer1List);
}

public ArrayList<String> removerAns2(int pos,ArrayList<String> answer2List){
    answer2List.remove(pos);
    return (answer2List);
}

public ArrayList<String> removerAns3(int pos,ArrayList<String> answer3List){
    answer3List.remove(pos);
    return (answer3List);
}

public void playMusicGame() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/onGame.wav");
        AudioInputStream audiIn = AudioSystem.getAudioInputStream(url);
    }
}

```

```

        // Get a sound clip resource.
        onGame = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        onGame.open(audioln);

        FloatControl gainControl = (FloatControl)
onGame.getControl(FloatControl.Type.MASTER_GAIN);

        gainControl.setValue(-40.0f);
        onGame.start();
        onGame.loop(5);
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

public void playMusicOnClickRight() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/right2.wav");
        AudioInputStream audioln = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        right = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        right.open(audioln);
        right.start();
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    }
}

```

```

    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

public void playMusicOnClickWrong() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/wrong2.wav");
        AudioInputStream audioln = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        wrong = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        wrong.open(audioln);
        wrong.start();
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}
}

```

File : startState.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.URL;
import javax.sound.sampled.*;

public class startState{

    private JFrame f = new JFrame("Idioms Quiz");
    private ImageIcon imgicon = new ImageIcon("Resources/Logo/logo2.png");
    private JLabel backGround = new JLabel();
    private JButton Go = new JButton();
    private boolean stopMusic = false;
    private Clip clip;
    private Clip click;

    public startState(){

        f.setIconImage(imgicon.getImage());
        f.setSize(805, 890);
        f.setLocation(500, 50);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout( null );
        playMusic();

        backGround.setLocation(0,0);
        backGround.setSize(850,850);
```

```
backGround.setIcon(new ImageIcon ("Resources/Background/startback7.jpg"));
```

```
Color c1 = new Color(112,0,1);
```

```
Go.setLocation ( 302, 500 );
```

```
Go.setSize (200,100);
```

```
Go.setBackground (c1);
```

```
Go.setIcon(new ImageIcon ("Resources/Picbutton/start.png"));
```

```
f.add(Go);
```

```
f.add(backGround);
```

```
f.setVisible(true);
```

```
Go.addActionListener(new ActionListener() {
```

```
    public void actionPerformed(ActionEvent evt) {
```

```
        playMusicOnClick();
```

```
        if (clip.isRunning()) clip.stop();
```

```
        //answer is Right
```

```
        new GUIGame();
```

```
        f.dispose();
```

```
    }
```

```
});
```

```
}
```

```
public void playMusic() {
```

```
    try {
```

```
        // Open an audio input stream.
```

```
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/start.wav");
```

```

    AudioInputStream audiIn = AudioSystem.getAudioInputStream(url);
    // Get a sound clip resource.
    clip = AudioSystem.getClip();
    // Open audio clip and load samples from the audio input stream.

    clip.open(audiIn);
    FloatControl gainControl = (FloatControl) clip.getControl(FloatControl.Type.MASTER_GAIN);
    gainControl.setValue(-40.0f);
    clip.start();
    clip.loop(5);
} catch (UnsupportedAudioFileException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (LineUnavailableException e) {
    e.printStackTrace();
}
}

public void playMusicOnClick() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/right.wav");
        AudioInputStream audiIn = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        click = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        click.open(audiIn);
    }
}

```



```
        click.start();  
    } catch (UnsupportedAudioFileException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    } catch (LineUnavailableException e) {  
        e.printStackTrace();  
    }  
}  
}
```

File : winState.java

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.URL;
import javax.sound.sampled.*;

public class winState{
    private JFrame f = new JFrame("Idioms Quiz");
    private ImageIcon imgicon = new ImageIcon("Resources/Logo/logo2.png");
    private JLabel backGround = new JLabel();
    private JButton play = new JButton();
    private JButton quit = new JButton();
    private Clip clip;
    private Clip playag;
    private Clip quitt;

    public winState(){

        f.setIconImage(imgicon.getImage());
        f.setSize(805, 890);
        f.setLocation(500, 50);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout( null );
        playMusic();
        backGround.setLocation(0,0);
        backGround.setSize(850,850);
```

```
backGround.setIcon(new ImageIcon ("Resources/Background/winback2.jpg"));
```

```
Color c1 = new Color(255,204,81);
```

```
play.setLocation ( 151, 500 );
```

```
play.setSize (200,100);
```

```
play.setBackground (c1);
```

```
play.setIcon(new ImageIcon ("Resources/Picbutton/play.png"));
```

```
Color c2 = new Color(255,48,49);
```

```
quit.setLocation ( 453, 500 );
```

```
quit.setSize (200,100);
```

```
quit.setBackground (c2);
```

```
quit.setIcon(new ImageIcon ("Resources/Picbutton/quit.png"));
```

```
f.add(play);
```

```
f.add(quit);
```

```
f.add(backGround);
```

```
f.setVisible(true);
```

```
play.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        playMusicClickplay();
        //answer is Right
        if (clip.isRunning()) clip.stop();
        new GUIGame();
        f.dispose();
    }
});
```

```

});

quit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        playMusicClickQuit();
        //answer is Right
        if (clip.isRunning()) clip.stop();
        System.exit(0);
    }
});
}

public void playMusic() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/congregation.wav");
        AudioInputStream audiIn = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        clip = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.

        clip.open(audiIn);
        FloatControl gainControl = (FloatControl) clip.getControl(FloatControl.Type.MASTER_GAIN);
        gainControl.setValue(-40.0f);
        clip.start();
        clip.loop(5);
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    }
}

```

```

    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

```

```

public void playMusicClickplay() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/play.wav");
        AudioInputStream audioln = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        playag = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        playag.open(audioln);
        playag.start();
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

```

```

public void playMusicClickQuit() {
    try {

```

```
// Open an audio input stream.
URL url = this.getClass().getClassLoader().getResource("Resources/Music/right.wav");
AudioInputStream audiIn = AudioSystem.getAudioInputStream(url);
// Get a sound clip resource.
quitt = AudioSystem.getClip();
// Open audio clip and load samples from the audio input stream.
quitt.open(audiIn);
quitt.start();
} catch (UnsupportedAudioFileException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} catch (LineUnavailableException e) {
    e.printStackTrace();
}
}
}
```

File : overState.java

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.io.*;
import java.net.URL;
import javax.sound.sampled.*;

public class overState{
    private JFrame f = new JFrame("Idioms Quiz");
    private ImageIcon imgicon = new ImageIcon("Resources/Logo/logo2.png");
    private JLabel backGround = new JLabel();
    private JButton play = new JButton();
    private JButton quit = new JButton();
    private Clip clip;
    private Clip playag;
    private Clip quitt;
    public overState(){

        f.setIconImage(imgicon.getImage());
        f.setSize(805, 890);
        f.setLocation(500, 50);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.setLayout( null );
        playMusic();

        backGround.setLocation(0,0);
        backGround.setSize(850,850);
    }
}

```

```
backGround.setIcon(new ImageIcon ("Resources/Background/overback.jpg"));
```

```
Color c1 = new Color(254,76,74);
play.setLocation ( 151, 500 );
play.setSize (200,100);
play.setBackground (c1);
play.setIcon(new ImageIcon ("Resources/Picbutton/play.png"));
```

```
Color c2 = new Color(64,19,13);
quit.setLocation ( 453, 500 );
quit.setSize (200,100);
quit.setBackground (c2);
quit.setIcon(new ImageIcon ("Resources/Picbutton/quit.png"));
```

```
f.add(play);
f.add(quit);
f.add(backGround);
f.setVisible(true);
```

```
play.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        playMusicClickplay();
        //answer is Right
        if (clip.isRunning()) clip.stop();
        new GUIGame();
        f.dispose();
    }
});
```



```

quit.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent evt) {
        playMusicClickQuit();
        //answer is Right
        if (clip.isRunning()) clip.stop();
        System.exit(0);
    }
});
}

public void playMusic() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/sadly.wav");
        AudioInputStream audioln = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        clip = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        clip.open(audioln);
        FloatControl gainControl = (FloatControl) clip.getControl(FloatControl.Type.MASTER_GAIN);
        gainControl.setValue(-40.0f);
        clip.start();
        clip.loop(5);
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```

    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

```

```

public void playMusicClickplay() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/play.wav");
        AudioInputStream audioln = AudioSystem.getAudioInputStream(url);
        // Get a sound clip resource.
        playag = AudioSystem.getClip();
        // Open audio clip and load samples from the audio input stream.
        playag.open(audioln);
        playag.start();
    } catch (UnsupportedAudioFileException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    } catch (LineUnavailableException e) {
        e.printStackTrace();
    }
}

```

```

public void playMusicClickQuit() {
    try {
        // Open an audio input stream.
        URL url = this.getClass().getClassLoader().getResource("Resources/Music/right.wav");

```

```
AudioInputStream audioln = AudioSystem.getAudioInputStream(url);  
// Get a sound clip resource.  
quitt = AudioSystem.getClip();  
// Open audio clip and load samples from the audio input stream.  
quitt.open(audioln);  
quitt.start();  
} catch (UnsupportedAudioFileException e) {  
    e.printStackTrace();  
} catch (IOException e) {  
    e.printStackTrace();  
} catch (LineUnavailableException e) {  
    e.printStackTrace();  
}  
}  
}
```