Project 1 Report

Rongsheng Qian 301449387

Overall:

1. Folder structure.

```
├── Q1.jar                // Question 1 executable file
├── Q2.jar                // Question 2 executable file
├── Sample                // Sample
│   ├── Q1
│   │   ├── audio1.wav
│   │   └── audio2.wav
│   └── Q2
│       ├── image1.tif
│       ├── image2.tif
│       └── image3.tif
├── file_chooser.java     // Main Class which implemented GUI and invoke two APIs from other two java.
├── read_tif.class
├── read_tif.java         // Offer API which show tif picture (don't run it directly)
├── read_wave.class
└── read_wave.java        // Offer API which show .wav file (don't run it directly)
```

2. Run

a. Using command

```
java file_chooser.java   // Q1, Q2 using same GUI
```

b. Using .jar executable file (Q1.jar, Q2.jar)

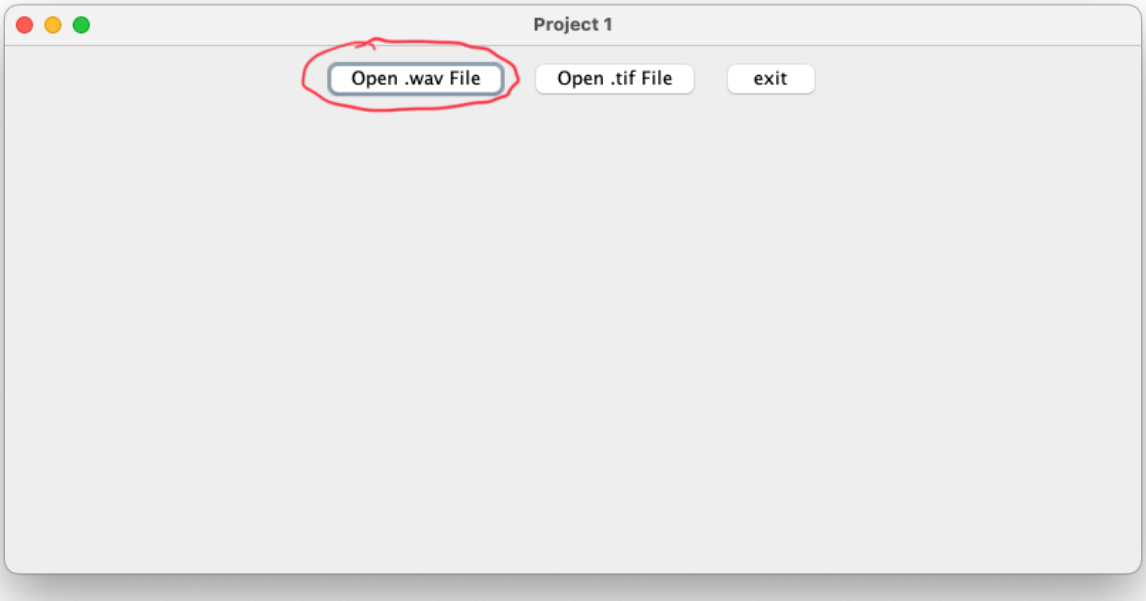
Q1 Explain:

1. Using API:

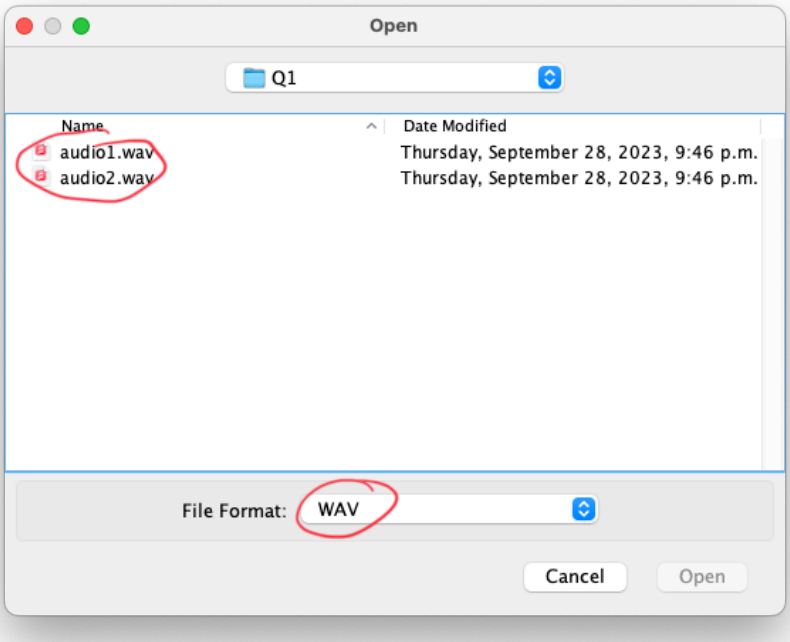
```java
// .wav read
import javax.sound.sampled.*;

// GUI
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
```

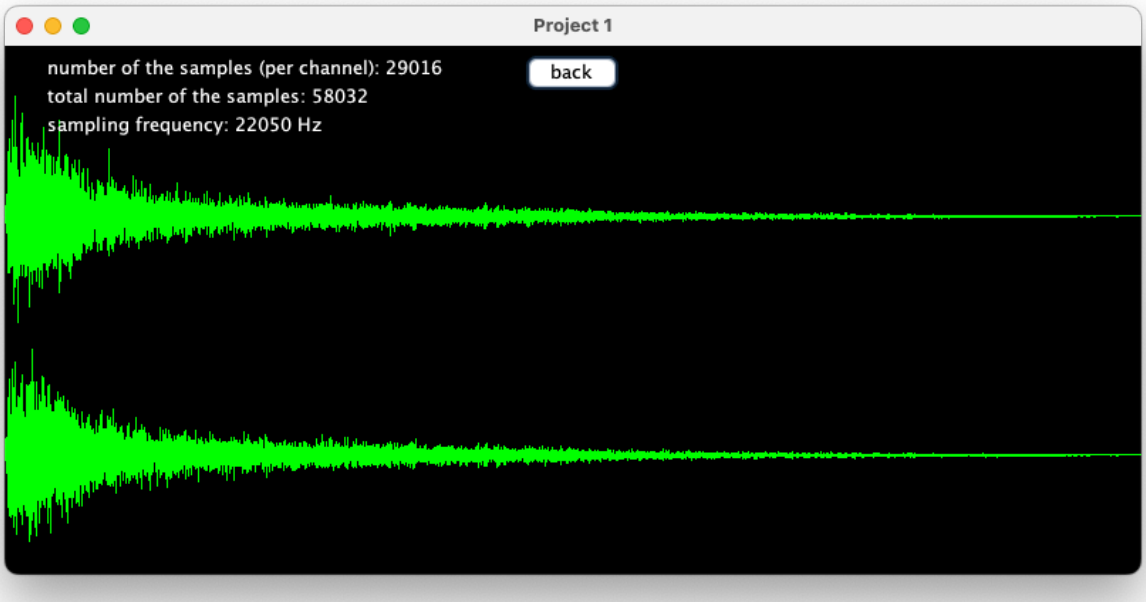
2. Screenshot:

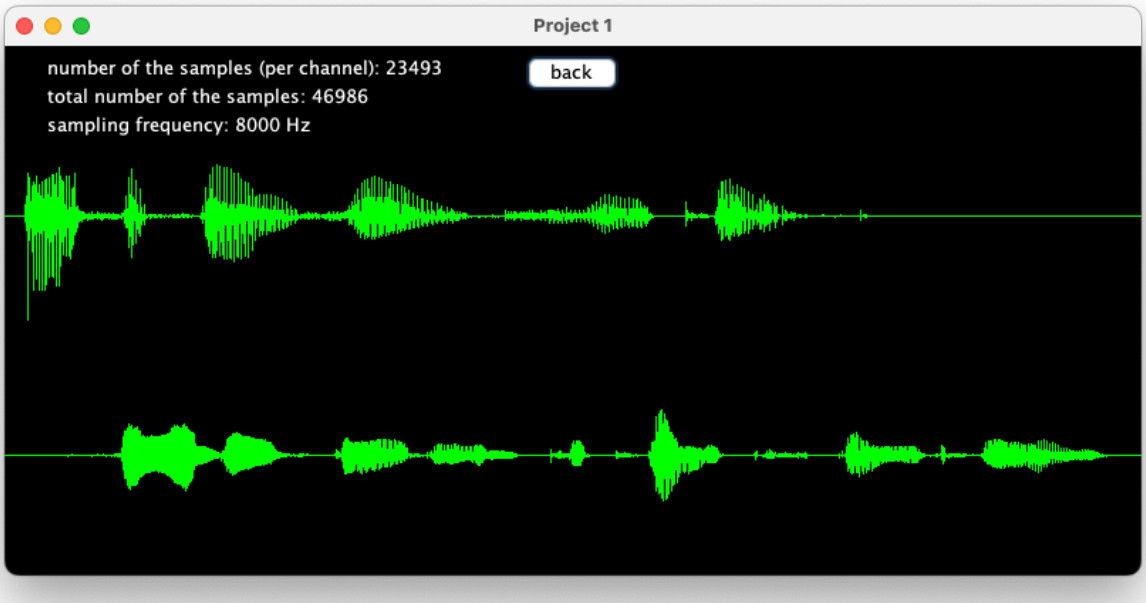


There is three button in starting page. (Open .wav, open .tif and exit). Exit will allow you to terminate the programe immediately.



This is open file dialogs which only allow you to choose .wav files. You can browser anywhere in your computer.





The back button allows you to go back the starting page and reopen other files (.wav and .tif)

Q2 Explain:

1. Implement

1.1.1 Write the parser by myself

I have just follower the guide written in the "Encyclopedia of Graphics File Formats" (Second ed.).

And get the coding_order and every information in IFD (tags, include tag_id, data_type, data_count and offset)

1.1.2 Problem of tif parser

Cuz of the flexibility formate of tif file which tag_data_offset (4 bytes) in tag structure can store either data or data offset.

The problem happened when the offset was storing data and the data was 2 bytes. In this case the useful infomation in tag_data_offset is only 2 bytes and the other 2 bytes will be the noise information which cause the result wrong.

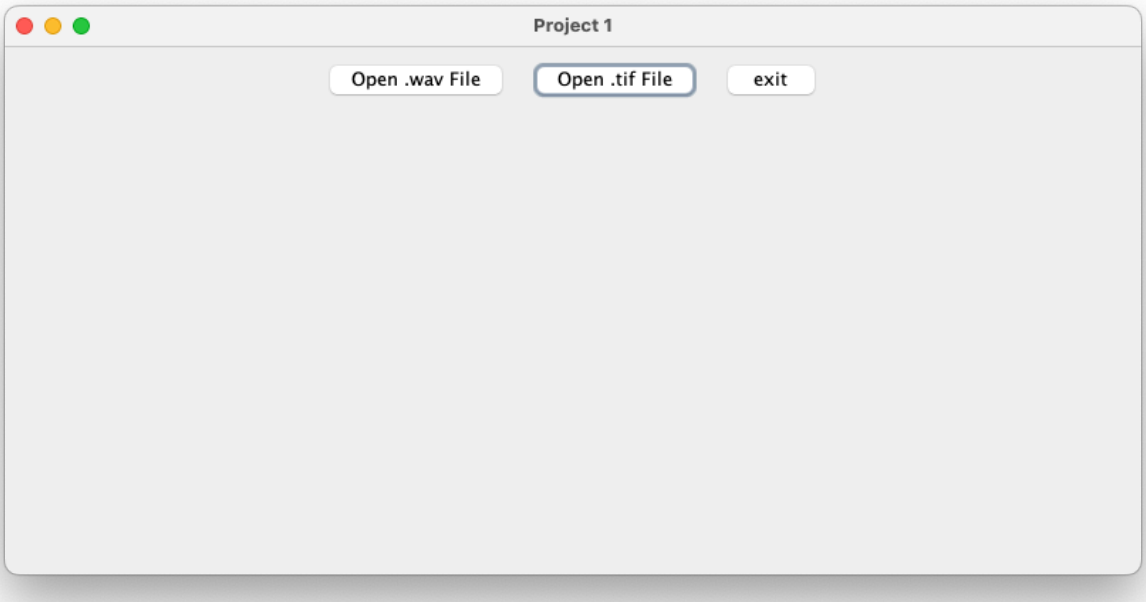
So there are too many cases needed to be considered and is hard to adapt for every tiff. **So I didn't put my parser in GUI.** I just write a main function and let it print out each tags info.

1.2 Using existing libraries for GUI

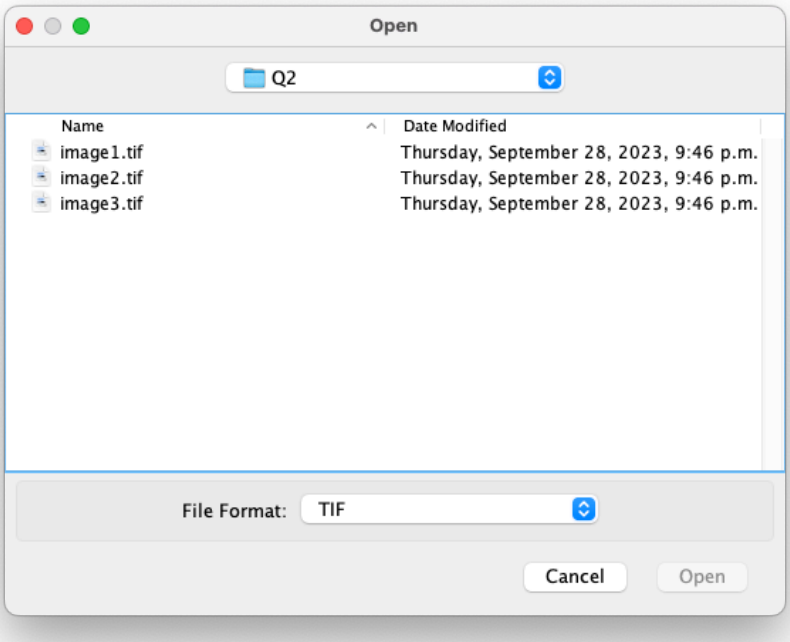
```
import java.awt.*;
import javax.imageio.ImageIO;

image = ImageIO.read(new File(imagePath));
g.drawImage(image, 0, 0, null);
```

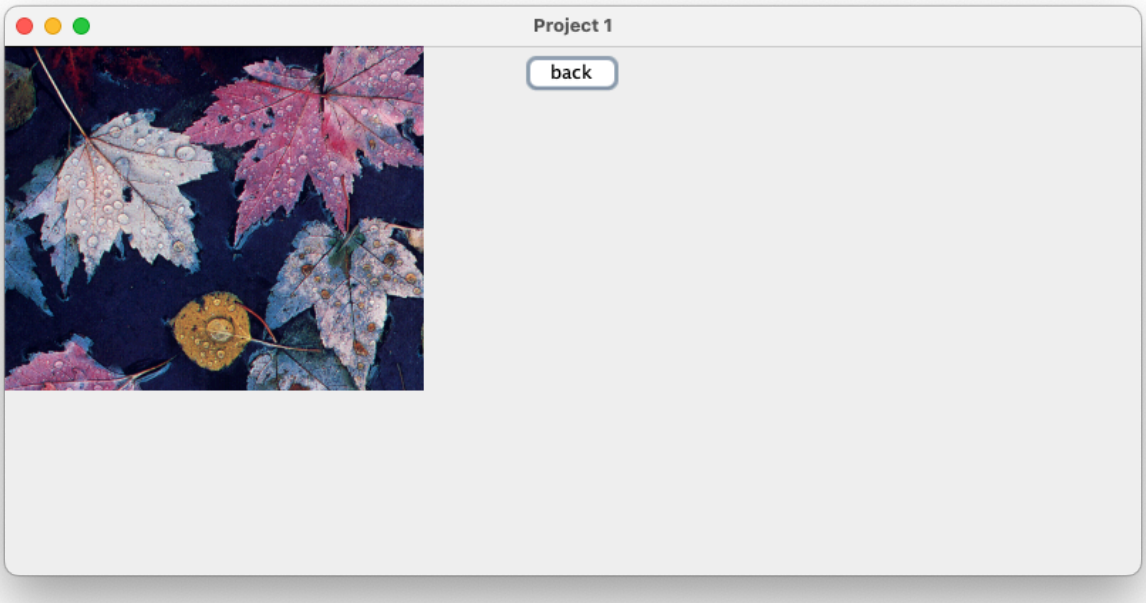
2. Screenshot:

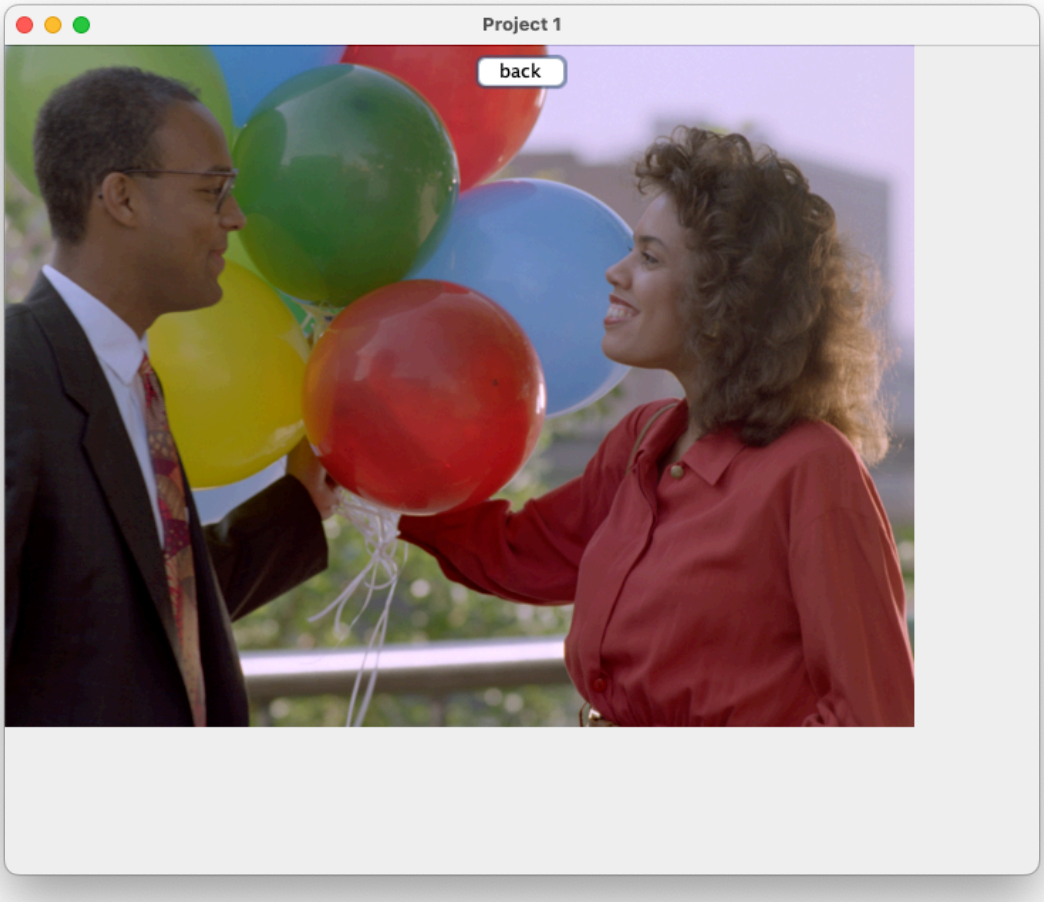


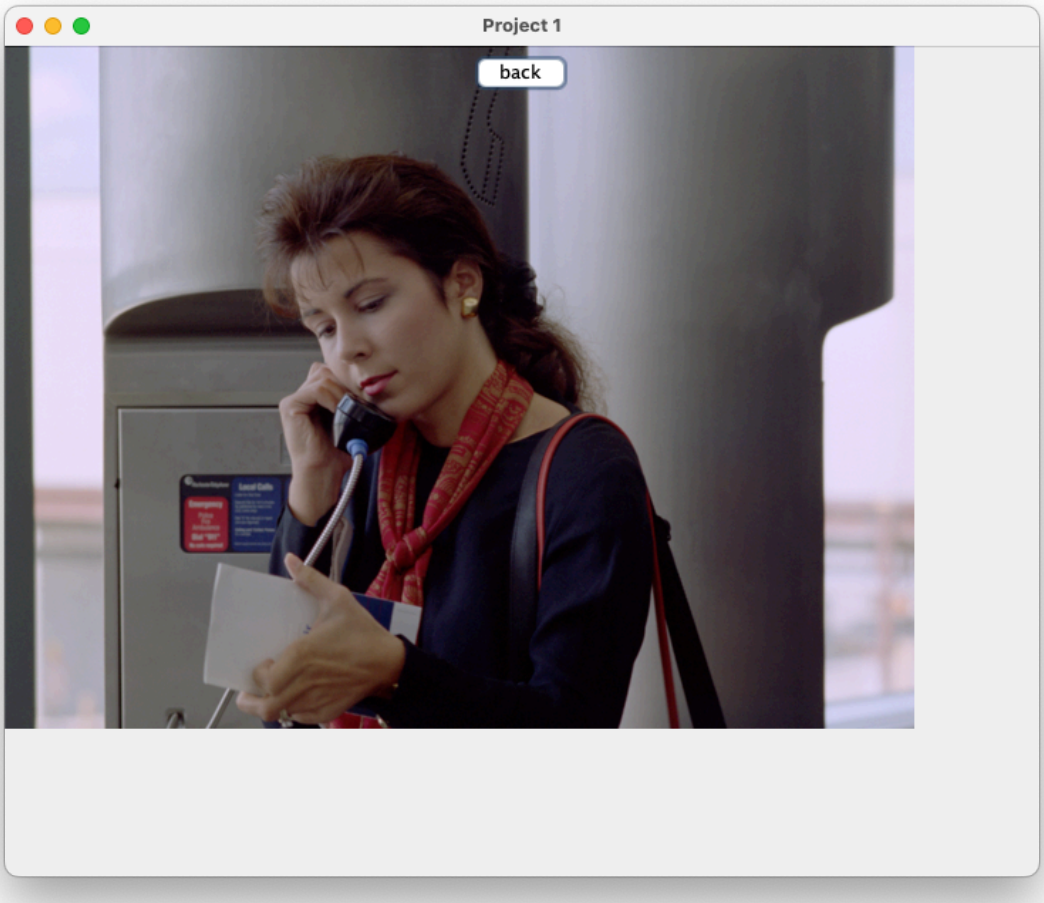
There is three button in starting page. (Open .wav, open .tif and exit). Exit will allow you to terminate the programe immediately.



This is open file dialogs which only allow you to choose .tif files. You can browser anywhere in your computer.


Project 1
back


Project 1
back



The back button allows you to go back the starting page and reopen other files (.wav and .tif)