CS2512 Authoring Report 2022-2023

Student Name: Rachel Lombard

Student Number: 121705031

Demonstrator’s Name: Dr John J. O'Mullane

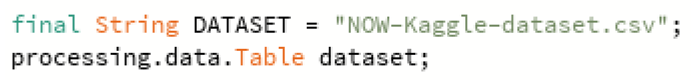
Assignment Number: 3

Submission Date: 11/12/2022

The goal of this project was to assign a personality trait when the user inputs their selection of preferred songs into the song selector. Once the user has given enough choices then the graphical user interface will show the user their personality trait and if they possess qualities of this trait or not. The data was taken from an excel data sheet which included a large number of volumes from ‘Now That’s What I Call Music’. A report of the code included to carry out this aim is below:

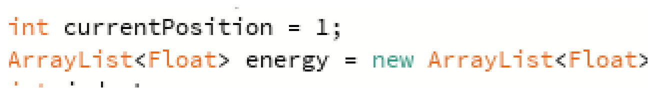
SelectASong -

Here in SelectASong the data set from the excel sheet is being called:



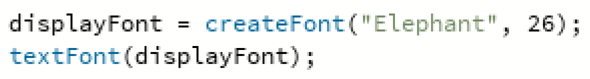
```
final String DATASET = "NOW-Kaggle-dataset.csv";
processing.data.Table dataset;
```

Created variable indexLen to stores the length of the energy array list to be used when checking the next volume:



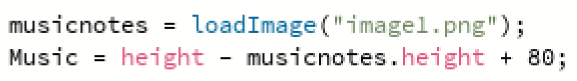
```
int currentPosition = 1;
ArrayList<Float> energy = new ArrayList<Float>
```

Edited the font type and size:



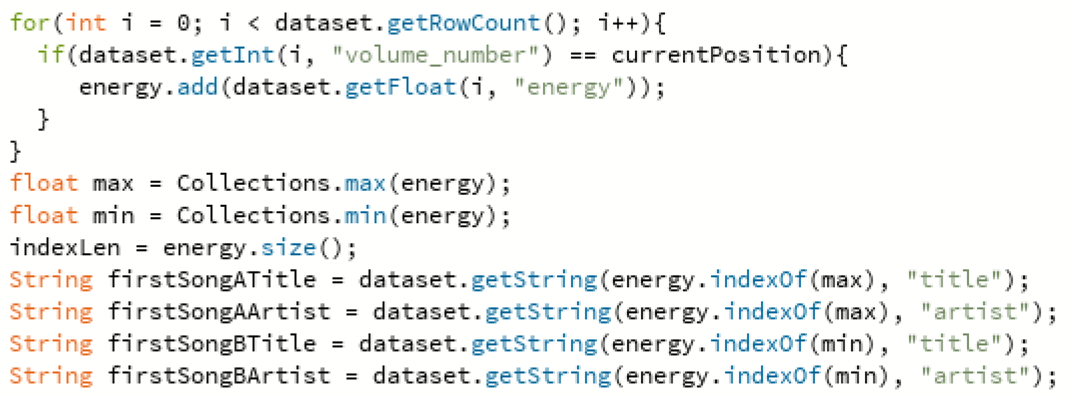
```
displayFont = createFont("Elephant", 26);
textFont(displayFont);
```

Added image so that it will be on the background of the graphical user interface:



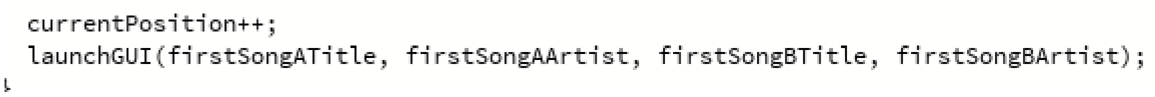
```
musicnotes = loadImage("image1.png");
Music = height - musicnotes.height + 80;
```

Created a for loop that went through all of the rows in the data set to add to the new energy array list, to hold the energy values. Found the max energy value and the lowest energy value and compared them. Used collections library to do this within the array list:



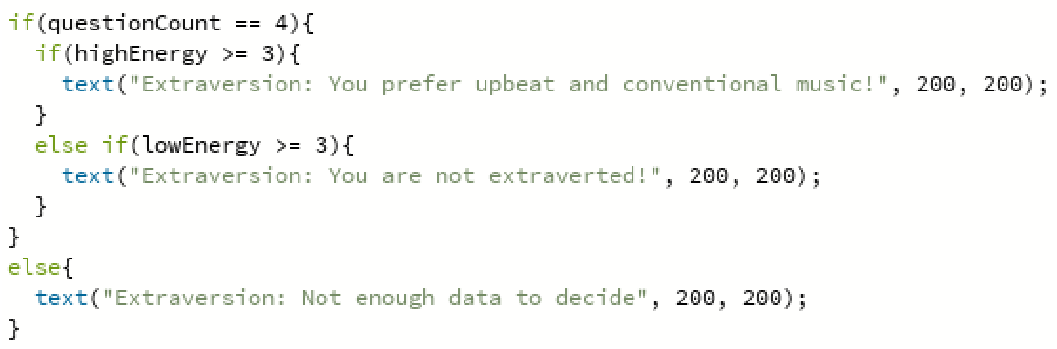
```
for(int i = 0; i < dataset.getRowCount(); i++){
  if(dataset.getInt(i, "volume_number") == currentPosition){
     energy.add(dataset.getFloat(i, "energy"));
  }
}
float max = Collections.max(energy);
float min = Collections.min(energy);
indexLen = energy.size();
String firstSongATitle = dataset.getString(energy.indexOf(max), "title");
String firstSongAArtist = dataset.getString(energy.indexOf(max), "artist");
String firstSongBTitle = dataset.getString(energy.indexOf(min), "title");
String firstSongBArtist = dataset.getString(energy.indexOf(min), "artist");
```

Stored variables such as Song A, Song B and incremented the position so that the operator goes to volume 2 and not stay at volume 1:



```
 currentPosition++;
 launchGUI(firstSongATitle, firstSongAArtist, firstSongBTitle, firstSongBArtist);
```

To include the personality data a conditional if statement was appropriate. After 4 choices and if 3 of those choices are 3 then show that the user is correlated with a specific personality trait. If they do not present the high energy choices then show that the user does not have that trait and not enough data on the interface will be shown if the user has not input enough choices. This will also remain on the interface until choices have been picked, e.g enough Song A choices:



```
if(questionCount == 4){
  if(highEnergy >= 3){
    text("Extraversion: You prefer upbeat and conventional music!", 200, 200);
  }
  else if(lowEnergy >= 3){
    text("Extraversion: You are not extraverted!", 200, 200);
  }
}
else{
  text("Extraversion: Not enough data to decide", 200, 200);
}
```

UI -

Similar to SelectASong code but begins at Volume 2. Add volume two to energy array list, found minimum and maximum of energy values. Indexlen was used to find minimum and maximum within the data set. Incremented position and added size of the next volume:

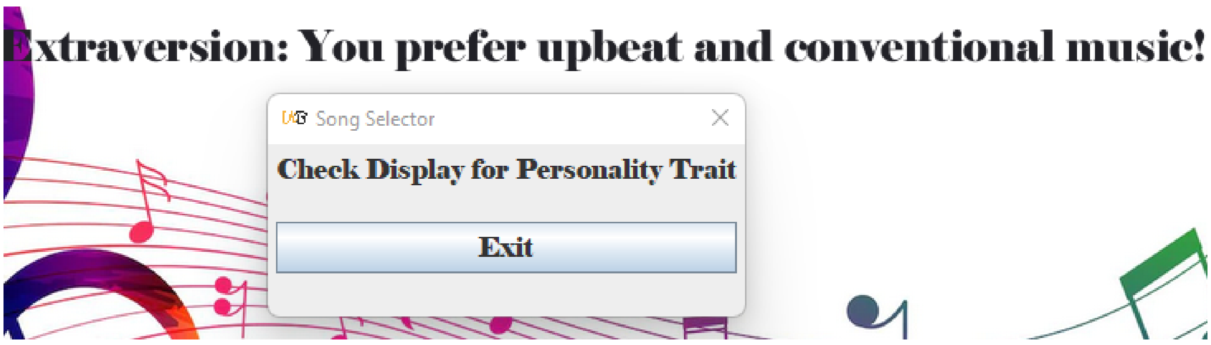
```
for(int i = 0; i < dataset.getRowCount(); i++){
 if(dataset.getInt(i, "volume_number") == currentPosition){
    energy.add(dataset.getFloat(i, "energy"));
 }
}
float max = Collections.max(energy);
float min = Collections.min(energy);
String songATitle = dataset.getString(energy.indexOf(max) + indexLen, "title");
String songAArtist = dataset.getString(energy.indexOf(max) + indexLen, "artist");
String songBTitle = dataset.getString(energy.indexOf(min) + indexLen, "title");
String songBArtist = dataset.getString(energy.indexOf(min) + indexLen, "artist");
form.getById("songA").setValue("Song A: " + songATitle + " : " + songAArtist);
form.getById("songB").setValue("Song B: " + songBTitle + " : " + songBArtist);
```

Cleared the energy list so the operator can focus on finding the max and min energy value on one volume at a time:

```
currentPosition++;
indexLen = indexLen + energy.size();
energy.clear();
```

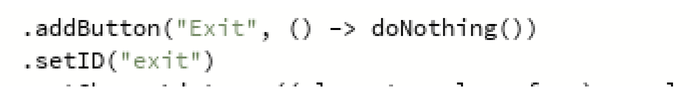
Once the trait has been calculated the user will receive a prompt menu for the user to check the interface:

```
.addLabel ("Check Display for Personality Trait")
//.setID("chooseOption")
```


Extraversion: You prefer upbeat and conventional music!

Check Display for Personality Trait

Also added an exit menu so that the user can exit the graphical user interface once their trait has been calculated:



```
.addButton("Exit", () -> doNothing())
.setID("exit")
```

References

Image URL: https://images.app.goo.gl/AVH4ZG2CLsUyjKeo6