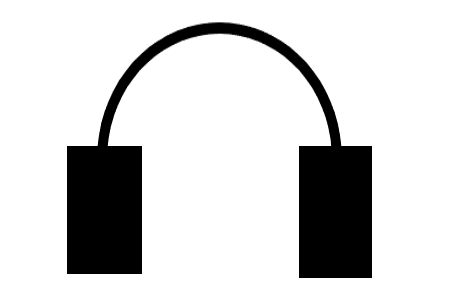
Zach Wolsborn and Ronnie Mohapatra

Ronnie and Zach’s Music Player

­­



Ronnie Mohapatra

Final Project Reflection

During this project we created a music player app that allows the user to create playlists, add and remove songs from the playlist, and add songs or playlists to a queue that holds the next songs to be played. We learned a lot about the File class in Java as well as the ins and outs of GUI. To aid us in our project we used YouTube to get .wav files for the songs, GitHub to collaborate on the project, SWT plugin to make the GUI, the Stack overflow and Oracle websites for help with errors and bugs within our project, and Spotify, which we modeled our app after. We were able to complete our whole project with all our intended functions and features.

One thing I learned during this project was setting the look and feel of a GUI program. I learned that different computers and operating systems have different installed look and feels. We had to experiment with different look and feels so Zach and I could have the same looking application. The default look and feel for a Mac is different than that for a Windows computer. Also, Zach and I did not want to have to install another Look and Feel just to synchronize our GUI programs. Fortunately, we both had a Look and Feel called Nimbus that worked well with our application and look the same on our computers.

The second thing I learned during this project was adding KeyListeners to a GUI program. A KeyListener is a listener that is triggered by pressing or holding down a key. We used these KeyListener to add keyboard shortcuts for playing, pausing, skipping, and going to the previous song. In order to actually implement these, the panel containing the listener must be focusable, but every other button must not be focusable. This was a major bug and took a long time to fix while working on our project.

One difficult concept was implementing multiple threads in our program. In our music player, I used multiple threads for auto-changing a song when it was over, and to constantly update the time slider and label. It was also a struggle to find some Java object that could play MP3 files instead of WAV files. This was a setback because WAV files are so much larger and much harder to compress.

One thing I wanted to add was a way for the application to store its current data and load it upon compilation, therefore every time you start the app, you still have the data you created from other times. We could have done this by writing to text file after creating a playlist or adding to the queue, and every time you run the code, the text file is read, and the data is restored. If there were more time, we could have figured out a way to do this.

I enjoyed the opportunity to learn about new concepts on our own. Being able to grow our coding knowledge was fun. Also, I found it exciting how vast Java is. Java has so many more untapped features than we have learned. Finding out about the different listeners and JComponents makes Java more fun to use. I also understand more real life applications of data structures such as Maps, ArrayLists, and Queues.