German Enigma Machine

My practical is a based off of German Enigma machine used during World War II for the German military to send encrypted messages. The actual Enigma machine had a series of rotors that would rotate but by how I have the letter map to connect encrypt this make rotation impossible. I can rotate in the encoding phase but when I go to decode it does not work. The way my program works is the user inputs a string to encode. The program changes the string into a data type that will work with the rotor. Once the string is in a list of Letters the list is mapped across ran through the rotors to come out with a output encrypted String. The repository that I have put it on is a GitHub.

<https://github.com/tucker19/GermanEnigmaMachine.git>

What I learned while working on this project was the actual makeup of the German Enigma machine and how to build one programmatically. I also needed to create a rotor that was a random assortment of Letters. I had to have the input letter run through these rotors and get a encoded letter. I also needed to be able to reverse this so I could decode. I could not get the rotors to actual rotate but I think I have figured that out. I think one of the most important things I learned was of how hard it is to reverse engineer this machine and see how much work it was done to break these codes back in World War II.

To run my program just compile within GHCI or GHC. Run the main function. Input if you want to encode or decode. Input your message to encode or decode, when you do this just type it in there is no need for “” by these will cause the program to throw an exception.