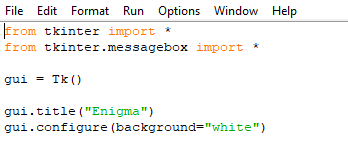
Logs and Development

#Problem 1

**Which language to choose?**

The first ever problem that I faced was which language am I going to use and I was stuck around two major high-level languages C# and python. As I am going to develop GUI-based software so the best language to use is C# but it’s complex coding and a structure made me to step back from it.   
Python is easy and fast to use. My main goal is to understand the cryptographical mechanism used in enigma machine. I can achieve this goal by using simple and understandable code for encryption and decryption of enigma. Easy code can make us understand how different components of enigma work for example plugboard, rotors, reflector keystroke and screen, how rotor rotates and how does it make encryption strong. I said no to classes in python programming language because I wanted our software to be as simple as possible but at the same time it does not affect our main goal.

I choose python over C#, our gui feature was tackled by using python library tkinter, our functions worked properly and our code was small and simple to understand.



#Problem 2

**Decryption was not working even after passing rotorlist [] array?**

Setting up the rotors only gave us result from plain to cipher text but we were not able to decrypt it back to plain text.

This issue was resolved by debugging, our rotorsettings was changing continuously and this continuous change effect our rotorlist [] as some chars were added and some were removed. This became a serious problem. After thinking one two hours, I came up with an idea of using two separate arrays. One array was used for the encryption and the second array was used to reset the whole array when decryption is called. This first array is called rotorlist [] and the second one is called rotorWorkinglist []. Now when decrypting, it first reset the values from the rotorlist [] and then it successfully decrypts the message.

#Problem 3

**Converting my command line interface working to graphical user interface?**

The third major problem that I faced was How am I going to convert my code into graphics, As I have already chosen tkinter as my python library but still I have to get a structure of how my software should look like.

Points that I noted as:

* Make logo
* Frame from 3 sliders that are connected to rotorsetting
* Using labels for everything
* Textbox for taking inputs from the user
* Enter button to run the input from the textbox to the algorithm
* Output box for displaying the result of encryption and decryption
* Clear text button that clears the output box

Outlining these points almost solved my problem, now I only had to work on the points one by one to get the finest output. Another important thing that I have to look was taking good care of errors and exceptions, for example when the fastest rotor is at Z it should shift to A automatically and at the same time middle rotor is also incremented by 1 alphabet.

Another problem that has to take care of, was the sequence of changing letter from one component to the next.

The sequence was

**Plugboard -> Rotor3 -> Rotor2 -> Rotor1 -> Reflector -> Rotor1 -> Rotor2 -> Rotor3 -> Plugboard**

In the first half the letter was changed from the alphabetlist to the plugboardlist / rotorlist / reflectorlist, but in the second half the letter was changed from the plug/rotor/reflector list to the alphabetlist.

Notes

Encryption:

sending msg should not be read by anyone else

During war encryption was important, now encryption is used by computers

Enigma German militarties1930 and throughout WW2

Keyboard -> 26

Lamboard -> 26

press one letter on keyboard lampboard outputs different letter

Sender needs enigma machine to encrypt it + rotor key

Receiver need enigma to decrypt it + rotor key

Understanding the circuit

#Mid

Rotors: it has 3 rotors, every rotor has 26 numbers

As it has 3 rotos single letter is changed 3 times

From A to D

From D to T

From T to H

that’s because of the wires which are scrambled

#Left

Reflector

#Right

Plugboard

Word is changed 7 to 9 times:

-->3 rotors x 2

-->1 reflector

-->Maybe plugboard x2

Logs

->Write Functions

->Link them

->Note Default values

->Take input

->Display Output

->Rotate Rotor Function

->Keep eye on sequence on character that changes through the whole algorithm

->Check Decryption for the encrypted text

->Transfer and link every component to graphical user interface

-> Add logo and title

->Add rotors slider and link them at backend

->Add Input Text-box and link it at the backend

->Add Buttons

* + Enter
  + Quit
  + Clear
  + Plugboard
  + Export

->Add Display Box and Display output over there

->Add Custom Plugboard and link it with the code that will change default plugboard with the custom one

->Add Export Feature to make a text file and dump output text in it + Add key of the rotor for decryption