Coursework Report

Piotr Handkowski
09005378@live.napier.ac.uk
Edinburgh Napier University - Web Technologies (SET08101)

1 Introduction

The aim of this report is to present the process of creating Cipher Me website. The motives and inspirations behind each ciphers will be explained, as well as the implementation in code. The structure and style of the website will also be presented giving an overview of the steps taken to create the website.

Cipher Me is a project that brings to the users three ciphers that have a completely different ways of encoding and decoding messages. The first two ciphers, Enigma and Polyalphabetic cipher, are implementations of existing ciphering techniques, used in the past. The last cipher, Zip cipher, is a new way of making a message unreadable that I came up with myself. They also range in difficulty. Zip cipher can be decoded using a pen and pager, whereas Enigma requires a lot more thinking and resources to crack it.

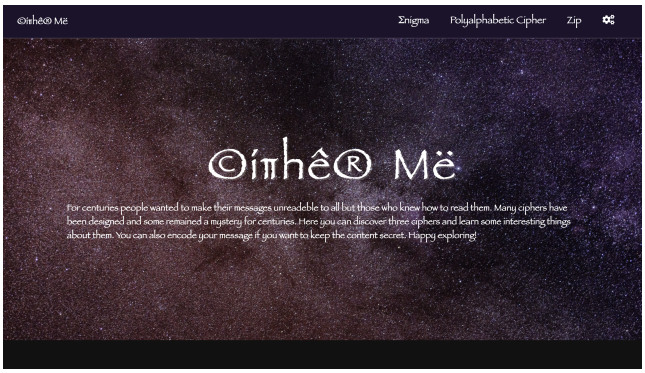

Figure 1: Cipher Me homepage

2 Visual design and implementation

The visual page design process started with a simple sketch of all planned pages and how the user will be able to navigate between them. The most obvious way of navigation seemed a navbar that would be included at the top of each page. This would ensure that the user would be able to navigate to any page from wherever they currently are. This has been implemented according to the initial plan. On the left hand side of the navbar there is a logo that takes the user to the homepage. On the right hand side there are links to each cipher, but also to the design document.

For the homepage, I wanted to keep it simple with just the logo and a short description of what the website has to offer. Since I do not have any design skills I created the logo using different characters to make them it look illegible at the first glance, additionally I used a mathematical sign pi to replace the letter 'p'. I have also used CSS transformations to make the logo and the description animated. When the page is opened all the characters fly into their position, rotating at the same time, to create the logo. The description then fades in.

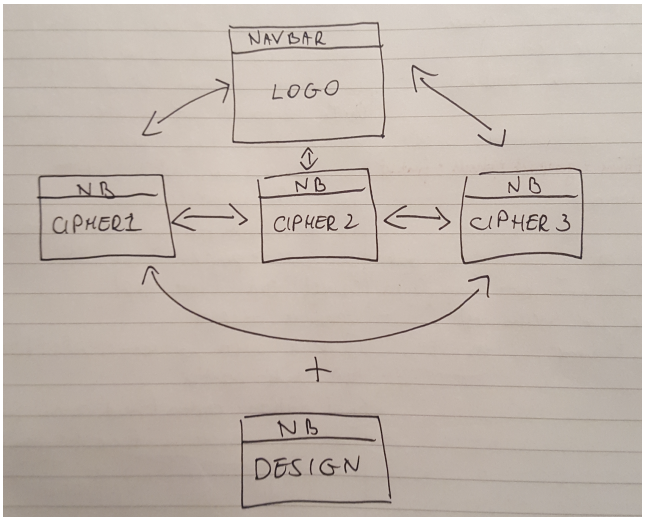

Figure 2: Structure design

The original design of each cipher page has also been fully implemented. With a cipher name at the top, some description of the cipher on the right and some controls to cipher/ decipher a message on the left. The split between the description side and the code/decode side has been implemented with a grid. The code/decode side takes 2 fractions of the container and the description side takes 1 fraction.

I have not decided on any particular colours, or themes during the initial design process, as the visual design is not my area of expertise. When I found a background photo (by Nathan Anderson at: https://unsplash.com/@nathananderson) I then started matching the colours and fonts to go with it. The background I used is a photo of stars in the dark space, therefore I used a dark purple colour for the navbar (1b122b, rgb(27,18,43)) and white font colour. After experimenting with the default CSS fonts I have decided to use the fantasy font, as it seems to go well with the rest of the components. I styled the rest of the components in a similar way, giving it the dark purple background colour with half opacity and white text. I have also added a fine opaque white border to the controls.

The cipher name and all the buttons in cipher pages are animated. The title slides from outside the screen and stops

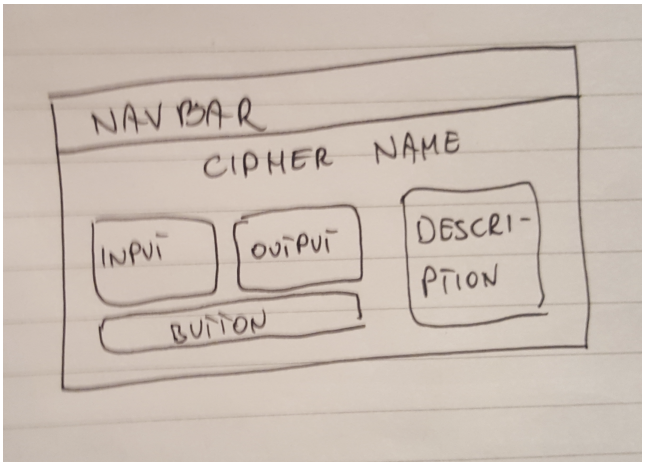

Figure 3: Cipher design

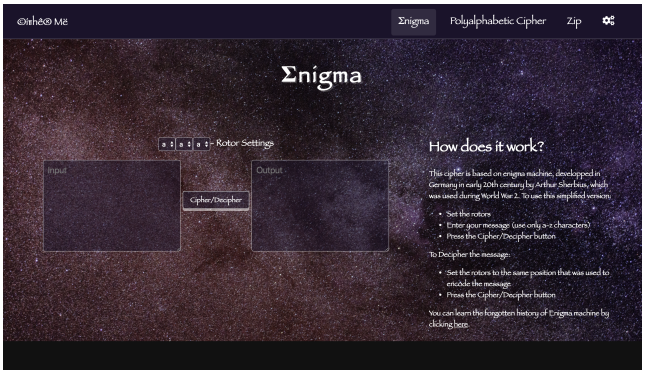

Figure 4: Cipher page implementation

in the middle. As for the buttons there is a hover and active effect which decreased the size of the bottom border and moves the position of the button down. This gives impression of the button being pressed. I have also decided to remove the active outline effect (blue outline), as it does not incorporate well with the theme.

Additionally the Enigma and Polyalphabetic ciphers have a link that opens a modal providing additional information about the ciphers, as well as links to YouTube videos (ref here?).

3 Ciphers and their implementations

I have decided to implement three ciphers with different levels of complexity and methods of encoding/decoding messages. Each cipher is created in a separate HTML file and has its own JavaScript file with the same name. Since they all use the same style only one CSS stylesheet has been created, named 'main.css'.

3.1 Enigma

The first cipher I decided to implement is Enigma. There are a few reasons backing up the choice. The main reason is it's history and the contributions to crack the code. Everyone heard about Alan Turing cracking the code, but not everyone knows about the contribution of Polish mathematicians who had to abandon their work due to lack of resources and Poland being occupied by Germany. As a Pole I decided to bring a little bit of the forgotten history [1] with my project. The user can find out about it by opening a modal with additional information. Another factor was the complexity of the machine. I wanted to push myself and implement in code a mechanism that was at the same time so simple and clever.

In order to learn more about the workings of the machine I watched a few videos, but the most efficient one turned out to be a video by Kevin [**?**] (Remember to check how to add ref). In his video he shows how to build a simple Enigma machine on a Pringles tube. He also provides a link to a pdf with all the elements that need to be cut out and put on the tube.

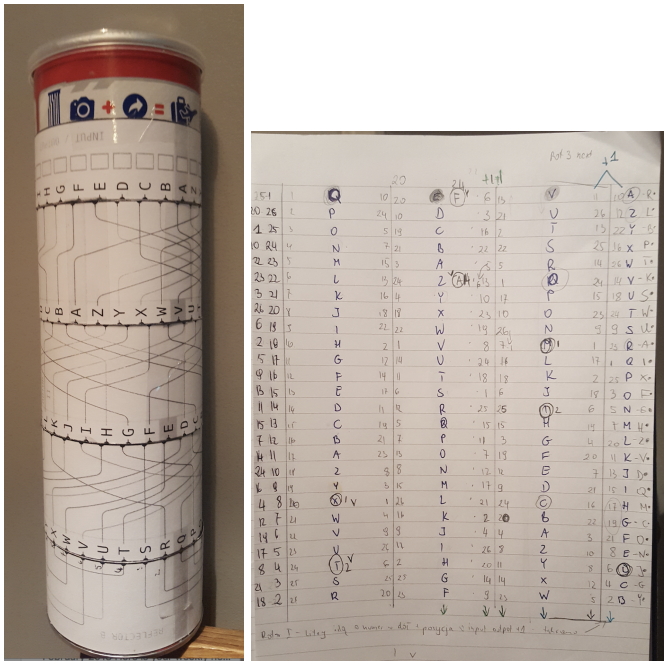

Figure 5: Enigma on Pringles tube and workings

This simple device helped me understand how the machine works and how the links between rotors change with each turn of a rotor. The second part of this challenging task was to implement this in JavaScript. I used an array of objects for each of the rotors, input/output and the reflector. Then I designed a method that encodes/decodes a single letter. The next task was to make the rotors move. This was quite challenging as a turn of a rotor not only affects this particular rotor, but also what the elements on the left and right point to. Therefore each rotor turn had to change data in three elements. When I completed this method I could use it to create another method that sets all three rotors when a message is to be encoded/decoded. With all those methods I was able to create a function and add it to a button that is taking the user input, encodes/decodes it and displays the result. I have tested the implementation against the Pringles tube device to see if I get the same result on both of them. I used different rotor setting to encode/decode short messages and the results I got were identical. To decode a message the user needs to know the exact set up of the three rotors, otherwise they can try each combination at a time out of 17576 possible combinations. The real Enigma machine allowed other settings, for example reflector settings, which

on it own give 100391791500 different combinations. The total number of different combinations is 1.06×10^{16} which contributed to Enigma's reputation.

3.2 Polyalphabetic Cipher

The second cipher I have implemented is Polyalphabetic cipher. It is one of the most known ciphers used for centuries. It is based on Caesar cipher, but it is more complicated. Caesar cipher uses one number to shift all letters in a message, when Polyalphabetic cipher uses a list of numbers, as long as the user wishes. This makes the cipher more difficult to crack, although it does not guarantee a safe encryption. The increased complexity, compared to Caesar cipher, was the main reason I decided to implement it, as I wanted to improve on my JavaScript and problem solving skills.

I have implemented it by creating an array of 26 alphabets all starting from a different letter.

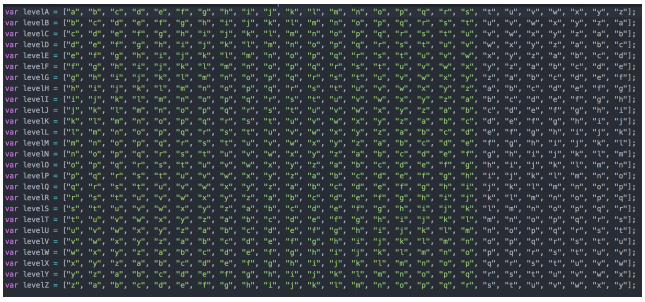

Figure 6: Poly alphabets

Depending on the key provided to decode a message, a group of alphabets are used. I created a method that changed the key, in this case a word, into a list of numbers. Subsequently those number indicated which alphabets were to be used in the process. For example a key 'cat' is changed to a list of numbers [2, 0, 19]. To encode a word 'hello' we could replace 'h' with a letter of the same index from the third alphabet(index[2]), letter 'e' with a letter from the first alphabet(index[0]), letter 'l' with a letter from the twentieth(index[19]), and we would repeat the process for the rest of the message. This would give us encoded word 'jeeno', as seen in the below example. To decode the message the index number of each letter is checked from the corresponding alphabet and a letter with the same index is returned from the base alphabet. I have implemented functions performing the above operations and tested it many times encoding and decoding messages. Without the key it is very hard to decode a message and the complexity increases with the length of the key. The user can learn a bit more about Caesar and Polyalphabetic ciphers by opening a modal with more information about them. There are also two links to YouTube videos which can aid understanding the encryption process of these ciphers and provide more information.

3.3 Zip Cipher

Having implemented two well known ciphers I wanted to try and create one of my own. The ciphering method I came up with rearranges letters in the message making it look like gibberish. The message is split into two messages and then zipped together character by character.

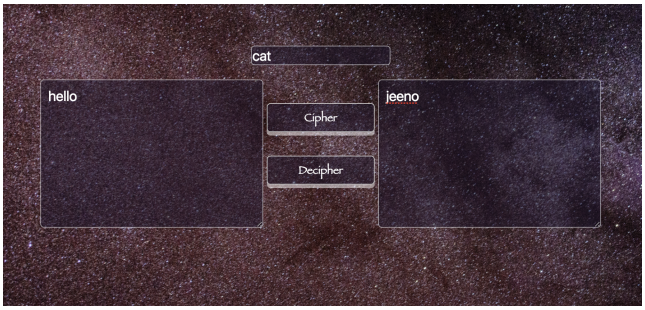

Figure 7: Polyalphabetic cipher example

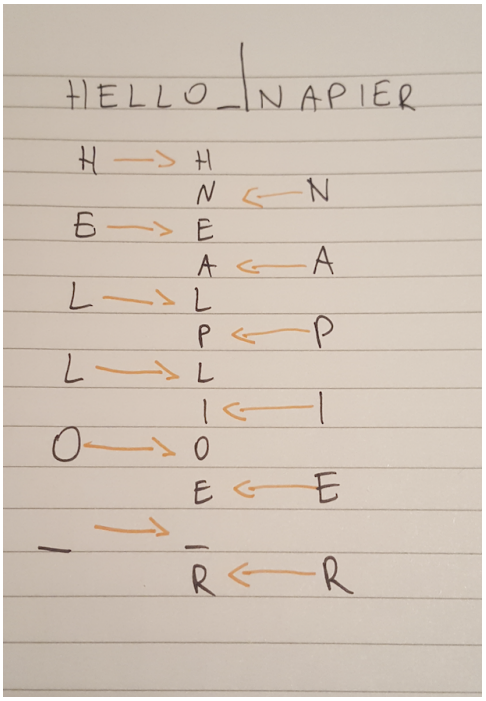

Figure 8: Zip cipher

Since it can be easily noticed that reading the odd letters first and then the even ones can reveal the original message the Zip cipher reverses the strings before zipping them. This makes the encoded message look less like real words. Although the method is quite simple it cannot be cracked with frequency analysis like Caesar, or Polyalphabetic ciphers. It requires the knowledge of how the letters were swapped in the message. When it comes to JavaScript implementation this was the easiest cipher with only 4 simple functions (divide, reverseString, zip and unzip).

Figure 9: Zip cipher example

4 Critical evaluation

This project implements three ciphers using only HTML, CSS and JavaScript. No other external libraries, templates or frameworks were used. The project includes 'index.html', 'design.html' files, as well as HTML file for each cipher implemented. Each cipher page contains a description of the cipher, input area to enter the message button(s) to cipher/decipher messages and an output area.

Although all the requirements for this project have been met, there is some things that could be improved if more time was allowed for this task:

- the navbar could be made responsive to display a hamburger icon on small screens that expands when clicked
- Enigma machine could be improved by allowing more set up to be made
- perhaps the theme of the project could be better implemented to make it look more modern

5 Personal evaluation

Despite small things that I think could be improved, for example the theme, I am quite happy with the finished product. Having some experience in web development and using libraries like Bootstrap, I had to take a step back and implement everything using only HTML, CSS and JavaScript. This made me feel more confident using those technologies. The biggest challenge was definitely implementing the workings of Enigma. Having worked on this challenge made me realize that implementing algorithms and finding solutions to different problems is the thing I enjoy the most about coding. This has also made me appreciate how ingenious the creators of Enigma must have been and how smart were those who unraveled it. I have also learned about one of the oldest ciphers known and how it can be cracked by frequency analysis. Another technology I have learned during this project is the LaTeX tool. It was quite discouraging at the beginning since I had to learn it from scratch, but now I can see the benefits of using it to make your report look more professional. I am already writing another report using LaTeX and I think I will be using it for every written assignment from now on.

To summarize I feel that completing this project was overall a positive experience with a lot of new things I have learned along the way.

6 Conclusion

References

[1] T. Sale, "The breaking of enigma by the polish mathematicians."