

MSc Software Development

An exploration of Cryptanalysis Learning Software through
the Zodiac Cryptograms

Rachel J. McRae

DEPT. OF COMPUTER AND INFORMATION SCIENCES
UNIVERSITY OF STRATHCLYDE

This dissertation was submitted in part fulfilment
of requirements for the degree of
MSc Software Development

University of Strathclyde, Glasgow

August 19, 2019

DECLARATION

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself. Following normal academic conventions, I have made due acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to place a copy of the dissertation in a publicly available archive.

(please tick) Yes [☒] No [☐]

I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is 18,533.

I confirm that I wish this to be assessed as a Type 1 2 ③ 4 5

Dissertation (please circle)

Signature:

A McRae

Date: 19/08/2019

()

Abstract

This dissertation explores the use of educational software within the scope of university-level cryptography and cryptanalysis. To thoroughly investigate the effectiveness of the design of cryptography and cryptanalysis learning tools - the methodology and implementation of 'Zodiac Cryptogram', an educational cryptography tool is detailed and evaluated. Through evaluation results, the Zodiac Cryptogram Tool displayed a notable advantage over the use of similar cryptanalysis learning resources.

Contents

| | |
|---|-------------|
| Abstract | ii |
| List of Figures | vi |
| List of Tables | viii |
| Preface/Acknowledgements | x |
| 1 Introduction | 2 |
| 1.1 Classical Ciphers and Cryptanalysis | 2 |
| 1.2 A Look at the Zodiac Killer Cryptograms | 2 |
| 1.3 Project Aims and Objectives | 3 |
| 1.3.1 Aim | 4 |
| 1.3.2 Objectives | 4 |
| 2 Literature Review | 5 |
| 2.1 An Introduction to Ciphers and Cryptanalysis | 5 |
| 2.1.1 Letter Frequency Analysis | 8 |
| 2.1.2 N-gram Analysis | 9 |
| 2.1.3 Chi Square Statistic | 9 |
| 2.1.4 Index of Coincidence | 10 |
| 2.1.5 Kasiski Examination | 11 |
| 2.2 Cryptography and Cryptanalysis in Higher Education | 11 |
| 2.3 Comparison of Existing Cryptanalysis Tools | 13 |
| 2.3.1 Educational Cryptanalysis & Cryptographic Visualisation Tools . | 13 |

Contents

| | | |
|----------|---|-----------|
| 2.3.1.1 | CrypTool 1 & 2 | 13 |
| 2.3.2 | Crypto-Tutor | 15 |
| 2.3.3 | VIGvisual | 16 |
| 2.3.4 | CET: Cryptographic Education Tool | 17 |
| 2.3.5 | Table of Comparison | 18 |
| 2.3.6 | Cryptanalysis Tools: Conclusion | 19 |
| 2.4 | Visualisation & Interaction in Learning Software | 21 |
| 3 | Project Management & Requirements | 23 |
| 3.1 | Project Management | 23 |
| 3.2 | Requirements | 25 |
| 3.2.1 | Requirements Gathering | 25 |
| 3.2.2 | Requirements from Discussions | 25 |
| 3.2.3 | Questionnaire & Results | 25 |
| 3.2.4 | Use Case Diagram | 28 |
| 3.2.5 | User Stories | 28 |
| 3.3 | Project Management & Requirements Conclusion | 30 |
| 4 | Software Design | 31 |
| 4.1 | Architecture | 31 |
| 4.1.1 | Cross-Platform Desktop Application GUI Frameworks | 31 |
| 4.1.2 | Qt5 Framework | 33 |
| 4.1.2.1 | Cross-platform | 33 |
| 4.1.2.2 | User Interfaces | 33 |
| 4.1.3 | Model-View-Controller | 34 |
| 4.2 | Class Design | 37 |
| 4.3 | User Interface Design | 38 |
| 4.3.1 | User Interface Design Principles and Choices | 38 |
| 4.3.2 | Prototyping | 38 |
| 4.4 | Software Design Conclusion | 40 |

| | | |
|----------|--|-----------|
| 5 | Implementation | 41 |
| 5.1 | Implementation Introduction | 41 |
| 5.1.1 | Programming Language & Coding Standard | 41 |
| 5.1.2 | JetBrains Pycharm IDE | 41 |
| 5.1.3 | Library Use | 42 |
| 5.1.3.1 | NumPy | 42 |
| 5.1.3.2 | Matplotlib | 42 |
| 5.1.4 | Analysis | 43 |
| 5.1.4.1 | Frequency | 43 |
| 5.1.4.2 | Ngram | 44 |
| 5.1.4.3 | Chi Squared | 45 |
| 5.1.4.4 | Index of Coincidence | 46 |
| 5.1.5 | Ciphers | 46 |
| 5.1.6 | Pages | 48 |
| 5.1.7 | Visualisation Implementation | 51 |
| 5.1.7.1 | Cipher Disk | 51 |
| 5.1.7.2 | Frequency Chart | 53 |
| 5.1.7.3 | Ngram Display | 54 |
| 5.1.7.4 | Chi Chart | 56 |
| 5.1.7.5 | Index of Coincidence Chart | 57 |
| 5.1.7.6 | Replace Text | 57 |
| 5.1.8 | Testing | 59 |
| 5.1.8.1 | Ad-hoc Manual Testing | 59 |
| 5.1.8.2 | Unit Testing | 59 |
| 5.1.8.3 | GUI Testing | 60 |
| 5.1.9 | Deployment | 60 |
| 5.2 | Implementation Conclusion | 60 |
| 6 | Evaluation | 62 |
| 6.1 | Evaluation Design | 62 |
| 6.1.1 | Comparative Experiment | 62 |

Contents

| | | |
|----------|--|------------|
| 6.1.1.1 | Analysis of Comparative Evaluation | 63 |
| 6.1.2 | Observation | 65 |
| 6.1.3 | Questionnaire | 66 |
| 6.1.4 | Interview Comments | 68 |
| 6.1.5 | Heuristic Evaluation | 74 |
| 6.1.5.1 | Heuristic Evaluation Summary of Results | 74 |
| 6.1.5.2 | Heuristic Evaluation Conclusion | 78 |
| 6.2 | Evaluation Conclusion | 78 |
| 7 | Conclusion & Recommendations for Future Work | 79 |
| 7.1 | Conclusion Introduction | 79 |
| 7.2 | Fulfilment of Research Questions & Project Aim | 79 |
| 7.3 | Recommendations for Future Work | 81 |
| A | Full Requirements Gathering Questionnaire | 83 |
| B | Use Case Diagram | 94 |
| C | User Stories | 95 |
| C.1 | Implemented | 95 |
| C.2 | Selection of Unimplemented | 97 |
| D | Evaluation | 98 |
| | Bibliography | 103 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Zodiac Killer Cipher Z408, letter and symbol frequencies | 3 |
| 2.1 | Image of cipher disk used by Confederate Army | 6 |
| 2.2 | Tabula Recta | 7 |
| 2.3 | Expected Index of Coincidence for Listed Languages | 10 |
| 2.4 | Screenshot of CrypTool 2 being used for frequency analysis. | 14 |
| 2.5 | Figure of student grade increase between students final grades | 15 |
| 2.6 | Screenshot of Z-408 template in CrypTool 2. | 16 |
| 2.7 | Screenshot of the Demo section of the AES algorithm of Crypto-Tutor . | 17 |
| 2.8 | Screenshot of the Practice Mode in VIGVisual | 18 |
| 2.9 | Screenshot of SDES demonstration in CET | 19 |
| 3.1 | Example of Hybrid Waterfall/Agile Approach | 23 |
| 3.2 | Example of Hybrid Project Management Workflow | 24 |
| 3.3 | Example result from Questionnaire | 26 |
| 3.4 | Selection of Cryptanalysis User Stories of the Zodiac Cryptogram Tool . | 30 |
| 4.1 | High-level view of MVC design of Zodiac Cryptogram | 35 |
| 4.2 | High-level view of MVC design of Zodiac Cryptogram | 36 |
| 4.3 | Prototype 1 User Interface for Zodiac section | 39 |
| 4.4 | Prototype 2 User Interface for Learn Caesar page | 39 |
| 5.1 | Letter Frequencies based on data from Google Books. | 44 |
| 5.3 | Class Structure for Ciphers | 48 |

List of Figures

| | | |
|------|--|-----|
| 5.4 | Multiple UI files in Qt Designer combined to create UI at runtime . . . | 50 |
| 5.5 | Code of LearnSimpleSub Page | 51 |
| 5.6 | Rotating cipher disk as shown on the Learn Caesar page | 52 |
| 5.7 | Example of Frequency Visualisation using section of a speech by President Jimmy Carter | 53 |
| 5.8 | Code snippet of plot() from 'FreqChart' which sets up and plots the chart | 54 |
| 5.9 | Ngrams displayed to help work out possible replacements | 55 |
| 5.10 | Code snippet of plot() from 'FreqChart' which sets up and plots the chart | 56 |
| 5.11 | Index of Coincidence chart | 57 |
| 5.12 | Replace text feature | 58 |
| D.1 | Full results from Evaluation Questionnaire | 103 |

List of Tables

| | | |
|-----|---|----|
| 2.1 | Table of Comparison between Cryptographic Tools | 20 |
| 6.1 | Table of Evaluation Results | 64 |

Preface/Acknowledgements

I would like to express my gratitude to my dissertation advisor Dr Rosanne English of the Department of Computer and Information Sciences at the University of Strathclyde. Dr English provided a considerable amount of expertise and advised me well throughout this project. I would also like to acknowledge the moral support from my partner Lukasz, my good friend Andrea and most of all, my mother for giving me the extra encouragement when I needed it most.

Chapter 1

Introduction

1.1 Classical Ciphers and Cryptanalysis

Cryptography, the art of writing in code, has been used for centuries by those requiring secrecy when sending and receiving messages. Its counterpart, cryptanalysis is the method through which these codes are broken. As cryptography has developed, so too have the cryptanalysis methods and tools used to unlock their secrets.

In the modern-day, more advanced cryptography is widely used in computing applications and with the ever-growing use of online banking, private messaging and online shopping there is a high demand for security and confidentiality across computer networks. As such, a solid understanding of cryptography and its applications has become an essential skill for those working in Computer Sciences.

1.2 A Look at the Zodiac Killer Cryptograms

In the 1960s and 70s a serial killer calling himself 'The Zodiac' was active in Northern California. To this day, his identity has not been uncovered by the police. He gained notoriety in part due to his communications with the police, press and local citizens - sending letters, including encrypted messages, which detailed his crimes and potential future crimes. In total he is known to have murdered five people and injured two, although he claimed to have murdered thirty-seven, this has not been confirmed. There are eighteen letters considered to be sent from the Zodiac, but once again, not all are

confirmed to be from the killer himself (zodiackiller.com, 2019).

The Z 408 cipher was the first and only cipher created by the Zodiac Killer which was considered solves. Donald and Bettye Harden, two amateur cryptographers, revealed its solution a week after it was published in newspapers. Bettye uncovered two "cribs" which helped to the couple to find the solution, one of which being 'kill', as the note was from the Zodiac Killer there was a high likelihood he would mention this either in the form of his alias or in talking about his acts (Bauer, 2017). Cribs are words that are likely to occur within a certain text, for example, the likelihood of 'attack' being used in wartime correspondence Bauer (1997)(McRae, 2019) . The cipher type used by the Zodiac killer was a homophonic substitution cipher, which used multiple letters and symbols to represent one alphabetical character, as illustrated in Figure 1.1.

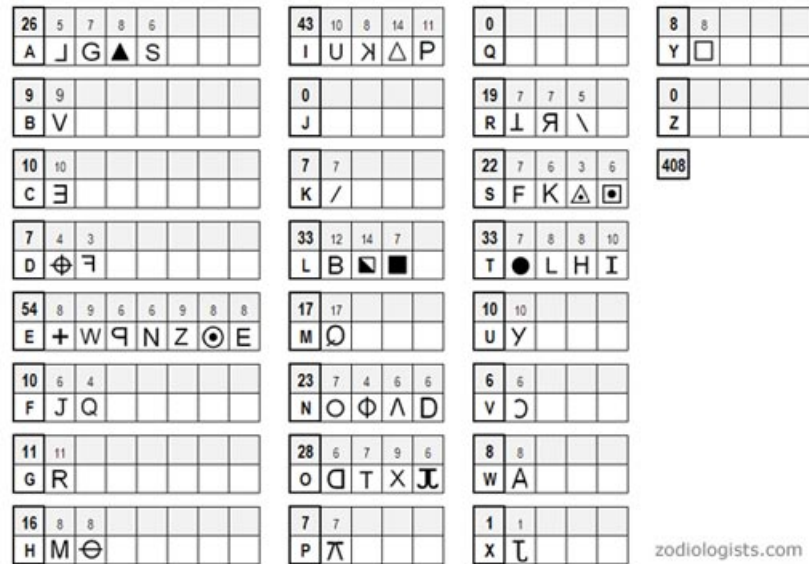


Figure 1.1: Zodiac Killer Cipher Z408, letter and symbol frequencies. (www.zodiologists.com, 2009)

1.3 Project Aims and Objectives

This project aims to explore cryptanalysis software that can interactively demonstrate the methods used to encrypt and decrypt ciphers. As the tool is intended to be a learning aid for cryptanalysis, the natural assumption is that the tool would be of most

Chapter 1. Introduction

use to those studying cryptography. Cryptography is most commonly taught in-depth in university courses within subjects such as Computer Science and Mathematics. Hence, the target audience would be University level students enrolled in STEM courses. The use of the Zodiac ciphers as a theme within the tool is indented to give an added level of intrigue through the use of a real-world cipher which remains unsolved. As such, the resulting tool was named 'Zodiac Cryptogram'.

It is not expected that any of the unsolved Zodiac Killer ciphers be solved within the scope of this project. Numerous knowledgeable researchers have tried and failed previously, additionally there is evidence, pointed out by Juzek (2018), which suggests the ciphers may have no meaningful plain text to reveal - potentially being fake pseudo-ciphers which cannot be solved.

1.3.1 Aim

This project hopes to fulfil the aim of;

Develop an educational cryptanalysis tool which aids the learning process for students and can be used to analyse a selection of ciphers interactively.

1.3.2 Objectives

To achieve this aim the following objectives have been created;

1. Explore how existing software is used to teach cryptography and cryptanalysis at a university level.
2. Investigate design elements which make these tools effective or ineffective at aiding student learning.
3. Develop an e-learning tool to aid university students in learning classical cryptography and cryptanalysis.
4. Evaluate the created software in terms of ease of use, effectiveness as a learning tool and its overall usability.

Chapter 2

Literature Review

This literature review begins by introducing a selection of the most common cipher types and cryptanalysis tools. Once a background has been established, the review examines cryptanalysis software tools which were created for educational cryptographic purposes. The following section considers the literature surrounding these tools and the evaluations conducted to assess their educational value within cryptography and cryptanalysis.

2.1 An Introduction to Ciphers and Cryptanalysis

One of the most famous examples is the Caesar Cipher - also known as the Caesar Shift Cipher. Named after the Roman Emperor Julius Caesar who supposedly used the cipher when communicating with his generals around 49-49 BC, as described by Singh (1999). It works by replacing each letter in the plaintext with another letter several places up or down in the alphabet.

When discussing ciphers, we often talk about plaintext and ciphertext. The plaintext is the message as it is written originally before being encrypted and ciphertext is the resulting message once the plaintext has been encrypted through some method.

When encrypting with a Caesar cipher, each letter is moved the same number of places, so there are no two letters which share the same value in the ciphertext. The shift would wrap around, so if a shift of 3 was used, the plaintext 'Z' would become

'C'. Hence the principle behind the cipher is often taught by using two rotating circles each with the alphabet laid out around them, known as cipher disks. An example of this can be seen in the figure below.



A Confederate cipher disk. (Courtesy of The Museum of the Confederacy, Richmond, Virginia. Photograph by Katherine Wetzel.)

Figure 2.1: Image of cipher disk used by Confederate Army (cryptome.org, 1997)

The Caesar Cipher falls under the category of monoalphabetic substitution (as it uses one alphabet to perform the encryption) and is a 'simple' substitution cipher. Simple substitution ciphers are based upon replacing each letter of the plaintext with another letter or symbol, resulting in the ciphertext. We consider substitution ciphers 'Classical' which often means they can be created without the use of mechanical machines or computers and can be achieved using pen and paper.

The Zodiac 408 cipher is considered a homophonic substitution cipher, Singh (2019) explains that a homophonic substitution can use multiple representations in ciphertext for one plaintext letter and can be applied so that it masks the expected letter frequencies in the plaintext. For example 'E' occurs at a rate of roughly 12% in the English language, so a cryptographer could use 12 different symbols or letters which

all independently represent the letter E.

Singh (1999, p. 45) also notes that polyalphabetic substitution ciphers were first known to be used circa the 1460s and attributed to Leon Battista Alberti. Polyalphabetic substitution ciphers are similar to monoalphabetic substitutions, but as the name suggested they use more than one 'alphabet' to encrypt the plaintext. This could be applied in a few different ways. In the Alberti Cipher, a Caesar shift with the addition of numbers was used, but the shift would change at various points throughout the ciphertext.

| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| B | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A |
| C | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B |
| D | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C |
| E | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D |
| F | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E |
| G | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F |
| H | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G |
| I | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H |
| J | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I |
| K | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J |
| L | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K |
| M | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L |
| N | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M |
| O | O | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N |
| P | P | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O |
| Q | Q | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P |
| R | R | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q |
| S | S | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
| T | T | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S |
| U | U | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T |
| V | V | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| W | W | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
| X | X | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W |
| Y | Y | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X |
| Z | Z | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y |

Figure 2.2: Vigenère ciphers are created using a Tabula Recta as shown in this image. The alphabet is laid out in each column and row starting from the position of the letter in the 'header' column and row.

This leads us on to the Vigenère cipher which built upon this principle of switching alphabets throughout the encryption process. First outlined in 1553 and unbroken until 1863 - it was described as *le chiffre indéchiffrable* according to Singh (1999, p. 63). The cipher is used by selecting a keyword or phrase which is repeated until its length matches that of the plaintext, which becomes known as the keystream (crypto.interactive-maths.com, 2019). The plaintext letter is located in the column header along the top and the corresponding keystream letter in the row header - their intersection within the table gives us the ciphertext letter. In the example above the plaintext letter is 'I' and the keystream letter is 'H' resulting in the ciphertext letter 'P'.

Transposition ciphers are also considered classical. A transposition cipher works by

keeping the letters the same as the original plaintext but changing their order until the message is unintelligible.

Classical ciphers were utilised mainly by spies and military over previous centuries. One of the more famous British examples of cryptography in a military setting was Bletchley Park, home to the British Government Code and Cypher School during World War II. Here they employed the use of machines to develop ciphers and to decrypt them. Most notably the use of Bombe machines developed by Alan Turing to break the German Enigma and Lorenz machine ciphers. This type of cryptography is not considered classical since it used rotor machines and plugboards to encrypt and decrypt. It does, however, lead us on to the development of modern cryptography.

Modern cryptography uses complex algorithms and computers to encrypt and decrypt. There are two main types being symmetric key and asymmetric key encryption. Symmetric key uses the same key to both encrypt and decrypt the ciphertext, whereas asymmetric key uses different keys to encrypt and decrypt.

These algorithms are exceptionally difficult to break even if you know exactly how they work. As such the most commonly used methods are widely known and even advertised, unlike classical ciphers of the past where knowing which cipher method was being used would lead to its decryption. This gives a number of advantages - firstly if the method is broken once it does not necessarily mean all previous or future applications of the method will be decrypted.

This brings us on to the methods used to decrypt these many ciphers through cryptanalysis. A lot of cryptanalysis is based upon comparing ciphertext with known features in the language the plaintext is thought to be in.

2.1.1 Letter Frequency Analysis

Frequency analysis is often the first port of call when examining a cipher. It is effective in breaking simple substitution ciphers as these do not hide the letter frequency of the language used. Frequency analysis is performed by counting all the letters or symbols used in the ciphertext and comparing this to the letter frequency of the language it is thought to be written in.

Additionally, it can also give clues to which language the plaintext was if it is not already known by comparing to letter frequencies of a range of languages. While a straight swap for each letter in order of frequency is unlikely to work, it points the cryptanalyst towards likely substitutions. Experimenting with switching out various popular letters in the ciphertext for those commonly used in English (or another language) can give us the first steps in solving the cipher. It should be noted that homophonic substitutions are more resistant to this method - since they often reuse letters and symbols for multiple letters, it disguises the natural frequency of the plaintext language.

2.1.2 N-gram Analysis

Technically frequency analysis as described above, falls under n-gram analysis and is known as monogram analysis. Additional types of n-grams extend this idea further. N-grams are sections of sequential text for n number of letters or words, i.e. monogram, bigram, trigram, quadgram and so on. They can also be made up of syllables, symbols and other collections but letters groups are what is commonly thought of when referring to n-grams. These items are often referred to as tokens. An example being common bigrams (two letters) in the English language such as; 'th', 'he', 'in' and 'er' (Mewhort, 2004). This can be used in breaking ciphertext - mainly substitution types - when the method of encryption does not hide the recurring groups of letters or words. If a plaintext containing several words containing 'th' was encrypted using a simple Caesar shift of 4 we might see a few occurrences of 'xl' appearing - leading us to suspect it to be one of the common bigrams.

2.1.3 Chi Square Statistic

The chi-square statistic is a method of comparing an expected measurement to an actual observed measurement. Typically there are two main ways of implementing a the Chi-Square Statistic; firstly, determining if there is a potential relationship between two sets of data and secondly, a fitness test which determines how closely the observed data matches the expected. Within cryptography the second is often used when attempting to determine what type of encryption has been used on a particular ciphertext.

When the two compared data sets are identical, the chi-square statistic will equal zero, the more difference between the two, the higher the resulting statistic will be. For example, the Chi-squared statistic of 'The message will be delivered at midnight' against an expected English text distribution is roughly 21.36. Typically a score of 150 or less indicated the text is in plain English. If this text was encrypted as a simple substitution, i.e. 'foz dzuuygz wjhh kz pzhjqzbzp yf djpaigof' gives a statistic of roughly 1888.84. Which indicates it is very far from the expected distribution of English.

2.1.4 Index of Coincidence

The Index of Coincidence of a piece of text can give a cryptanalyst hints as to its cipher type and the language of its plaintext if it is a simple substitution cipher. The I.o.C. is a measure of how likely it would be if you found two matching letters if you randomly selected them from a piece of text. In cryptanalysis by comparing the I.o.C for each letter it can be possible to determine the cipher type. For example, a Vigenère cipher would display a more even distribution I.o.C. across all letter types where as a simple substitution would be similar to English.

The I.o.C can be normalised by multiplying it by the number of letters in the alphabet used. Which in English would be 26. Figure 2.3 shows the normalised Expected Index of Coincidence for a selection of languages, as calculated by the United States National Security Agency in 1946.

| <i>Language</i> | <i>Expected I. C.</i> |
|------------------------|------------------------------|
| Random text | 1.00 |
| English | 1.73 |
| Russian | 1.77 |
| Italian | 1.93 |
| Spanish | 1.94 |
| Portuguese | 1.94 |
| French | 2.02 |
| German | 2.04 |

Figure 2.3: Expected Index of Coincidence for Listed Languages (NSA, 1946)

2.1.5 Kasiski Examination

This method accredited to Friedrich Kasiski in 1863 is a method of scrutinising polyalphabetic substitution ciphers which use a keyword. It is commonly used when attacking Vigenère ciphers. The reason for this is that the main weakness of a Vigenère cipher is that once you know the key length, the ciphertext can be split into a series of independent shift ciphers.

The method relies on n-grams within the ciphertext, the idea being that it is possible two bigrams or trigrams within the plaintext have been encrypted with the same key section at some point. This would result in a repeated bigram or trigram somewhere in the ciphertext. If one of these repeated ngrams is located it means the encryption of that repetition has potentially occurred at a multiple of the key length. To apply this as many n-grams that can be found in the ciphertext and their position, the distance between a selection of promising repeating n-grams is then compared with the hope of finding a common dividable number for the distances. If one is found you can assume every nth letter of the key length is encrypted via the same shift. These can then be split and analysed using other methods to crack the shift. The Kasiski Examination requires a fair amount of thought on the side of the cryptanalyst, requiring them to think critically about the possible key used, the more likely n-grams and the frequency of letters in the isolated shifts once the key length is found.

2.2 Cryptography and Cryptanalysis in Higher Education

In this section we should consider the need for software learning tools, specifically for cryptography and cryptanalysis within Computer Science education at a university level. Firstly, there is the perceived difficulty of cryptographic concepts presented at university level. As noted by Adamovic et al. (2018, p. 256) and Katz (2014, p. 14:1), students with weak mathematical backgrounds, especially those lacking in number theory, abstract algebra, probability, and statistics can become intimidated and struggle to grasp cryptographic systems when taught in university. This is likely in relation to the study of modern cryptography where most of the principles heavily rely on these

Chapter 2. Literature Review

concepts.

We should also consider what aspects of cryptography are commonly taught at a university level. Schembari (2007, p. 7) considered that cryptographic strength, encryption types and key management are core aspects of the Cryptography module taught at the University of Pennsylvania and teach both historic and modern encryption methods through experiential learning.

Schembari (2007) and Adamovic et al. (2018) discussed in the aforementioned papers cover both classical and modern encryption types in cryptography courses. It seems that even though classical cryptography is far too weak to be used for present day software applications it is still widely taught at university level. Spillman (2002, p. 2) describes why the study of classical ciphers is still important for computer science students;

”First, classical cryptology can motivate a student’s interest in contemporary cryptology. Second, classical cryptology can develop a student’s intuitive feel for the strengths and weaknesses of any cipher system. Third, classical cryptology can teach students the necessary discipline that they will need for the development and analysis of contemporary systems.”

By helping students to feel more comfortable with classical ciphers as a introduction to the subject it can give them drive to continue on to learning about modern cryptography.

Changji et al. (2009) state that there is a lack of educational software for information security teaching and students often find learning the subject through one method alone to ‘dull and dry’. Being able to provide a means to motivate and engaged students desire to learn could go a long way to helping them develop their understanding of the principles of cryptography. The processes behind cryptography can be difficult to grasp through text alone.

Adamovic et al. (2018, p. 257) outline the need for an interactive approach to cryptography learning as student evaluations in their cryptography course showed textbook theory was unsatisfactory and prior research from other authors, some of which are discussed in the next section of this literature review, showed statistically significant

grade increases from interactive learning tools.

2.3 Comparison of Existing Cryptanalysis Tools

2.3.1 Educational Cryptanalysis & Cryptographic Visualisation Tools

To address the issues stated in the previous sections a selection of cryptanalytic learning tools have been trailed in educational settings. In this section we will examine these past trails and their testing methods through the following criteria:

1. Type of Cryptanalysis taught
2. User Interface
3. Visualisation & Interaction
4. Evaluation Methods & Results

2.3.1.1 CrypTool 1 & 2

CrypTool is an open-source desktop software which is an e-learning platform for cryptography and cryptanalysis (Cryptool.org, 2019). CrypTool 2 builds upon its predecessor, CrypTool 1 by introducing more cryptographic types and analysis tools. Meaning both versions fall within our scope of educational applications. It is notably the most widely reviewed tool in terms of literature. CrypTool 2 is written using C#. They feature a wide selection of both classical and modern ciphers, which can be both used for encryption and decryption via the programs. The tool features animations which users can step through each stage of the encryption method, visualising the process at their desired speed. The CrypTool suite of products are open source and frequently updated, which has lead to them covering a wide range of cipher types.

CrypTool 2 has been used in several experiments to gauge its effectiveness as a cryptographic teaching tool in higher education.

Yang et al. (2011, p. 27) surveyed students who had used CrypTool as part of their learning and found that the visualisation element of the tool was "*invaluable*" in

Chapter 2. Literature Review

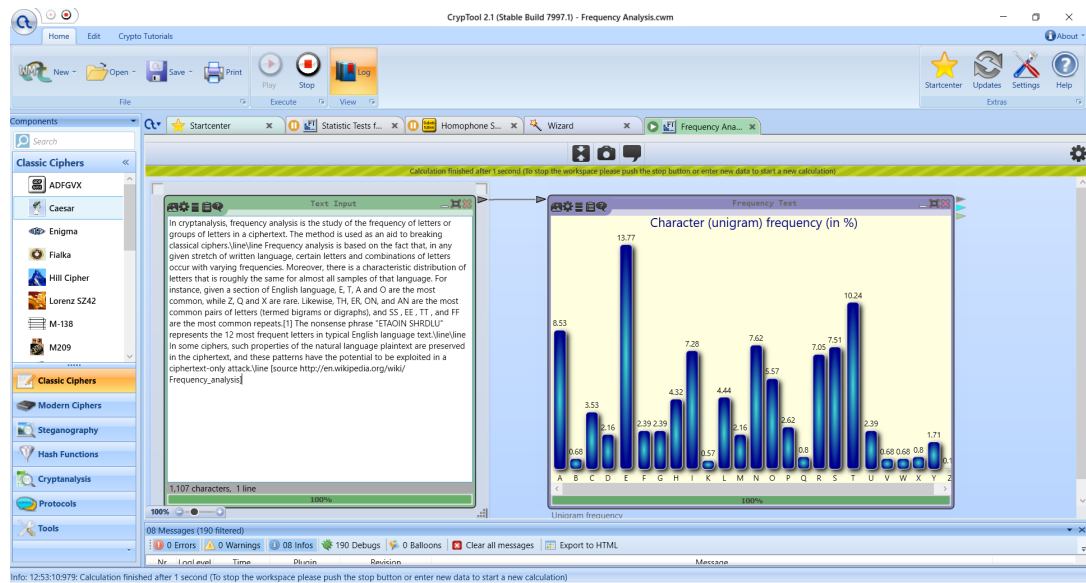


Figure 2.4: Screenshot of CrypTool 2 being used for frequency analysis.

demonstrating cryptographic techniques. However, specific quotes from these students are not listed in the paper - merely a general summary of comments made.

Interestingly, Adamovic et al. (2018) found a notable increase of student attendance and interest when introducing the use of CrypTool 2 to their Cryptography Course, on top of increased student grades. The grade increase observed can be seen in Figure 2.5. There is a notable improvement within the grades, with more students attaining grades above an 8. Adamovic et al. attribute this increase to their switch to a 'new interactive approach' with the inclusion of CrypTool. Potentially it could be that the interaction and visualisation are aiding student learning both by making the subject matter easier to understand, less intimidating and possibly less dry compared to the more traditional textbook and blackboard methods. If students feel more engaged with the material, it is a likely outcome that they would attain a better grade.

"Schuelerkrypto" or student crypto, is a German event aimed at engaging high school students in cryptography and cryptanalysis. During this event, students use CrypTool 1 and 2 to solve exercises which "...are designed as a special agent story i.e. a game-based approach to find the thieves of some famous old paintings." Hick et al. (2012).

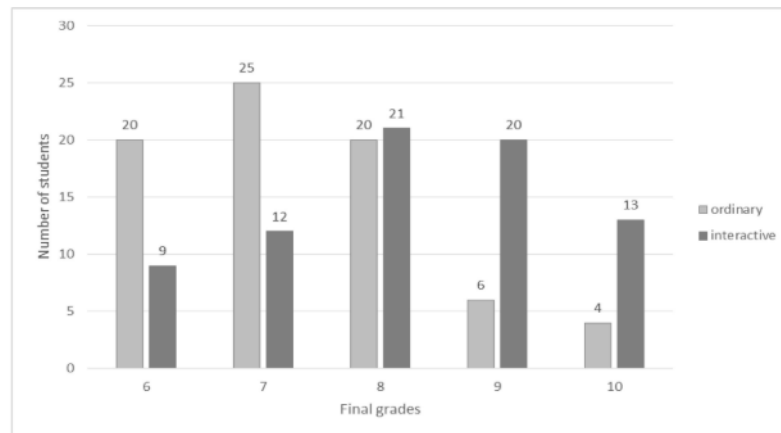


Figure 2.5: Figure of student grade increase between students final grades between using the interactive CrypTool 2 as part of their learning without. Adamovic et al. (2018, p. 261)

Curiously, the event organisers chose to include a narrative element to the exercises. Further investigation as to whether the narratives aided in the engagement and learning of the attendees would have been of interest. Hick et al. surveyed students (aged 14 - 19) who took part in the event 90% agreed that CrypTool was helpful as an e-learning tool and the same percentage agreed the event itself helped them to improve their knowledge in cryptography. This survey seems to suggest that e-learning tools, in this case, CrypTool, can help students to develop an understanding of cryptography. Hicks et al. do not mention how large their sample size of students was or use a control group who took part in exercises without CrypTool however, which prevents us from drawing a more robust opinion on the effectiveness of CrypTool as a cryptographic e-learning tool.

While the current stable build of CrypTool 2 does not feature the Zodiac ciphers a more recent 'Nightly Build' has a template for the Z-408. Templates in This can be analysed using the 'Homophonic Substitution Analyzer'

2.3.2 Crypto-Tutor

Luburi et al. (2016) outline Crypto-Tutor, a tool for teaching Created with Node.js and AngularJS. The program is split into several sections; the demo section allows

Chapter 2. Literature Review

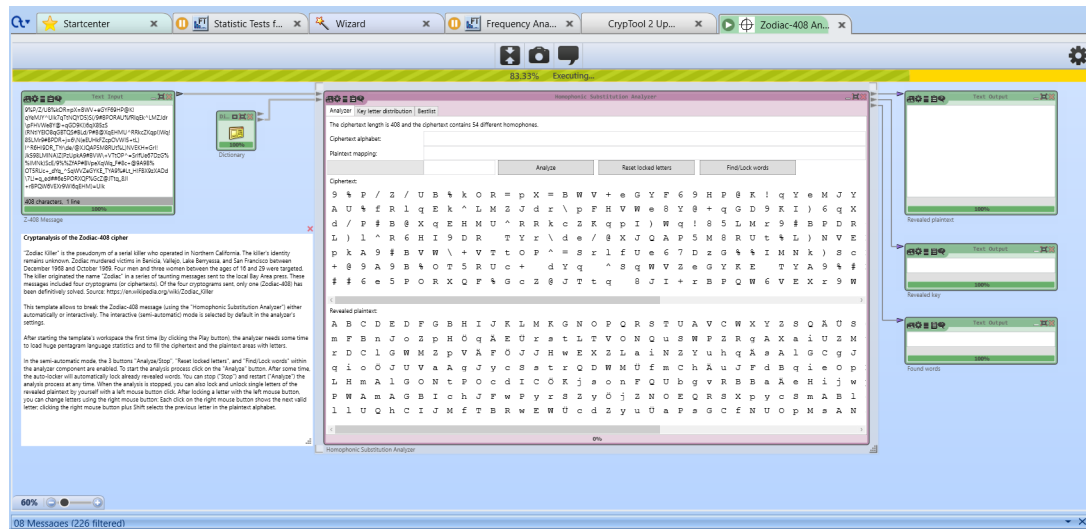


Figure 2.6: Screenshot of Z-408 template in CrypTool 2.

students to encrypt and decrypt text interactively. However, it does appear that this interactivity is limited to changing the plain and ciphertexts - this could be due to modern cryptography being harder to visualise. The tool also includes a quiz section to test students on what they have learned.

Luburi et al. (2016) found a slight increase of 20% in the average score of students they quizzed after studying with only Crypto-Tutor versus those who studied using other materials, concluding while it is only a slight increase it shows the software offers at least the same level of understanding. These results are promising; however, as the author acknowledges themselves, the quiz was rather short with a small number of participants.

2.3.3 VIGvisual

VIGvisual focuses on the Vigenère cipher, the use of Kasiski Analysis and the Index of Coincidence method to analyse the ciphers. The Kasiski method of breaking the Vigenère cipher by exploiting the repetition of the key is explained earlier in the introduction. The tool consists of three modes; demo, practice and attack. VIGvisual is part of a wider range of tools used to teach cryptography including; *SHAvvisual for the Secure Hash Algorithm*, *DESvisual for the DES cipher*, *AESvisual for the AES cipher*,

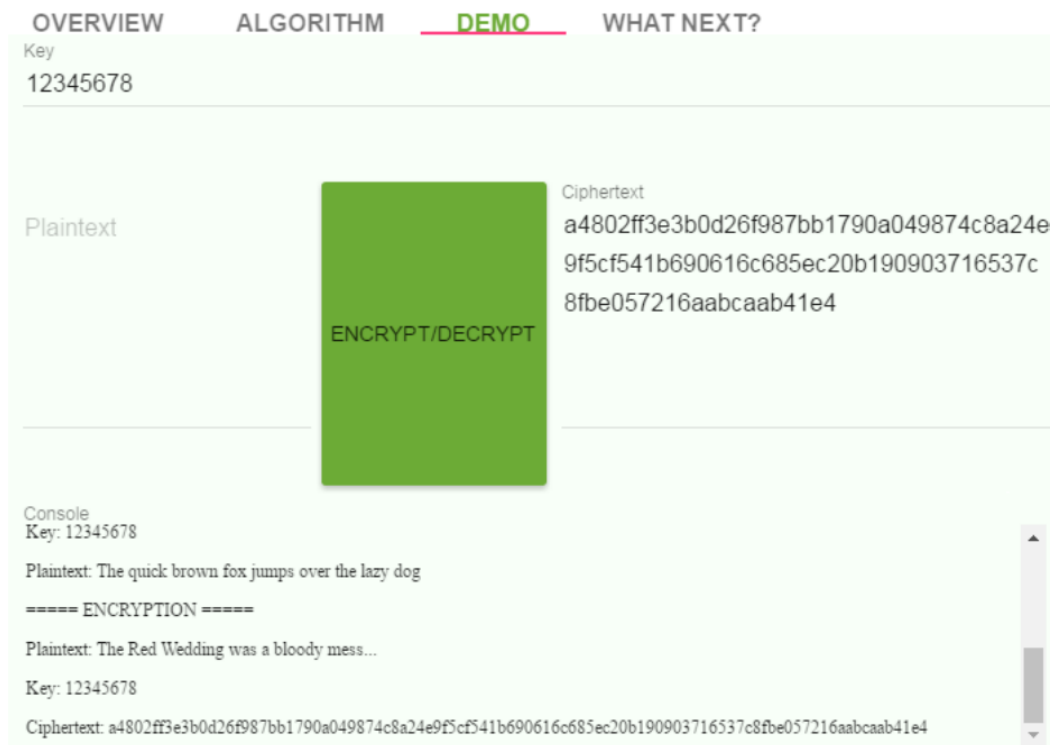


Figure 2.7: Screenshot of the Demo section of the AES algorithm of Crypto-Tutor. Luburi et al. (2016, p. 208)

RSAvisual for RSA cipher, and ECvisual for the elliptic curve-based ciphers.’ Li et al. (2015, p. 134). Each visualises a specific cipher type for educational purposes. The VIGvisual was used as part of an *Introduction to Cryptography* course at Michigan Technological University.

Li et al. (2015) surveyed 25 students who used VIGvisual; these students reported that the software overall both helped them to understand the cipher and enhanced the course. A comparison in test scores between two similar quizzes on the Vigenère cipher, issued before and after the introduction of VIGvisual, completed by 33 students revealed a significant increase in student’s scores in the second test.

2.3.4 CET: Cryptographic Education Tool

CET created as a visualisation-based teaching aid and demonstrates change case, simple cipher, RSA and SDDES.

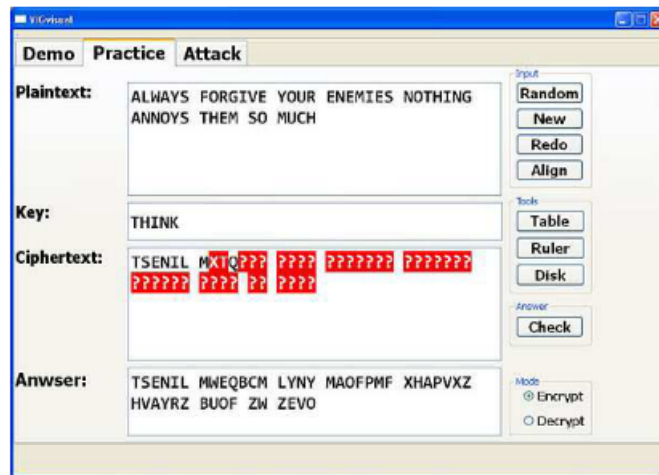


Figure 2.8: Screenshot of the Practice Mode in VIGVisual. Li et al. (2015, p. 130)

The tool was trailed during a semester in a Web Security class of both undergraduate and graduate students; questionnaires were used to gain student feedback on the tool. Of the fourteen students in the class, 100% agreed or strongly agreed that the demonstrations were helpful and that the tool helped them understand encryption algorithms Abuzaid et al. (2011).

2.3.5 Table of Comparison

To compare the reoccurring themes between the tools, a table of comparison was created, Table 2.1 can be seen on Page 20. There is a common idea between many of these tools where they are split into distinct sections — often giving an overview of the cipher, a more in-depth look at its inner workings, and a demonstration or interaction. Some go on to implement a fourth phase where users can create their own ciphers and apply the knowledge gained. This mirrors the 'Theory-Algorithm-Practice-Application' teaching mode for cryptography courses laid out by Yang et al. (2009), which the authors claim can yield better teaching results. The majority of these tools have not been update recently with the exception of CrypTool 2.

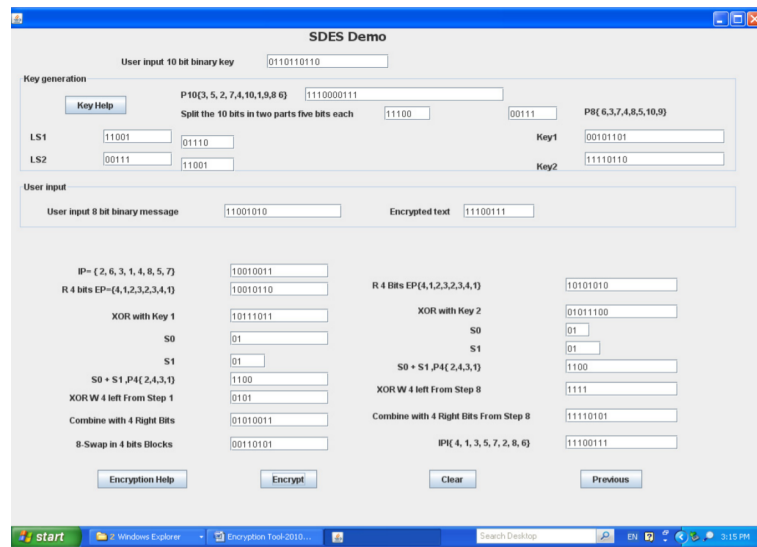


Figure 2.9: Screenshot of SDES demonstration in CET. Abuzaid et al. (2011, p. 196)

2.3.6 Cryptanalysis Tools: Conclusion

CrypTool and VIGvisual have shown a positive effect on the grades and learning of students who used them as part of their cryptographic learning in higher education. The main method of evaluation used by the various authors to draw their conclusions about the success of the tools was via qualitative surveys answered by student participants.

| Tool | Cryptography taught | Interaction | Visualisation | User Interface | User Interface - Student Comments | Evaluation through test results (based on Author comments) | Evaluation through opinion | Last Updated |
|---------------------------|--|--|--|---|---|--|--|---------------|
| CrypTool 1 & 2 | Wide range of classical and modern | Allows combining multiple methods of analysis, text inputs | Wide variety of animations of most ciphers | Uses a node based system which can be dragged and dropped | - | Significant improvement [1] | Students found the visual demonstrations enhanced their understanding. [2] | Daily updates |
| Crypto-tutor | Symmetric key, Public-key, RSA, ECC | Text inputs | Limited to static graphs | Split into Overview, Algorithm, Demo for each encryption type. Side bar for navigation. | - | Marginal improvement [3] | - | 3 years ago |
| VIGvisual | Vignère only | Text inputs, interaction with spinning shift disk, and slider, error checking of decryption attempts | Uses highlighted text in a step by step animation, graph for kasisk result, disk, table and slider representations | Split into Demo, Practice and Attack. | Some students found the table used to illustrate the Kasiski Test confusing [4] | Significant improvement [4] | Students found tool useful - some found in-depth understanding came from the blackboard but others preferred the faster and more visual method given by the tool [4] | 4 years ago |
| CET | Change case, simple cipher, RSA and SDES | Text inputs, error checking of decryption attempts | Step-by-step visualisation | Basic layout is split into key, encryption, decryption and demonstration for each category. | All students found the GUI user friendly [5] | - | All students agreed the tool helped them understand the ciphers and encryption algorithms taught via the tool [5] | - |

Table 2.1: Comparison of Tools mentioned above. Authors cited; [1] Adamovic et al. (2018) [2] Yang et al. (2011), [3] Luburi et al. (2016), [4] Li et al. (2015), [5] Abuzaid et al. (2011)

2.4 Visualisation & Interaction in Learning Software

'In sum, the "educational" aspect of educational software design is enhanced if a similar concept to designing classroom instruction is applied: information is presented, students are guided through the learning path, repeated practice is utilised for perfecting performance, and finally assessing students to establish whether or not learning has taken place. This does not mean that all these elements should be present at the same time in all educational software. More often than not educational software is used alongside teachers and other media to complete the instructional process, therefore, the elements are shared between them.' – *Pellone (1995, p. 73)*

This earlier work by Pellone (1995) has similar ideals to that of Yang et al. (2011) [check this isn't yang 2009], that the information is given in a guided manner, practised and then later tested to ensure learning. However, he does also note that as the software is often used alongside traditional teaching methods it is not always necessary to try combine the entire learning experience into one piece of software.

Cryptography often involves a lot of algorithms and complex processes. One of the barriers to understanding these processes is that they typically cannot be seen unless visualised in some manner. When discussing the design of apps to teach cryptography (Schweitzer and Baird, 2006) emphasised that high interactivity, consistency of representation between screens, minimal keyboard interaction, stand-alone descriptive text and algorithm representation (visualisation) were all important elements. Firstly, with regards to interactivity it seems this idea is somewhat supported in the comparison between tools with test scores having slightly less of an increase in Crypto-tutor, which also had the least amount of interaction and visualisation.

Li et al. (2015, p. 129) note that of the available cryptographic learning tools, most neglect to visualise the process and often exclude cryptanalysis. With Yang et al. (2011, p. 27) identifying visualisation as a key aspect which enhanced students learning of cryptography it would be of benefit to explore this further by including it in the design of this software. The notion of visualisation being a useful learning aid in computer

Chapter 2. Literature Review

science courses has been supported for a number of years by authors including; Fouh et al. (2012), Rößling et al. (2002) and Alhamdani (2016). While researching the use of algorithm visualizations in computer science education, Fouh et al. (2012, p. 110) found that at the time of their asking over 90% of CS educators had a strong positive view of the visualizations but only 10% used them frequently used them when teaching data structure courses.

Chapter 3

Project Management & Requirements

In this section, the project planning, methodology and requirements are laid out. As part of the research questions, it was determined that a cryptography-focused learning application should be designed, prototyped, developed, tested and evaluated in order to further explore the subject. In order to do this, it was important to select a method of project management that would keep development on track and carefully consider the requirements software such as this would have.

3.1 Project Management

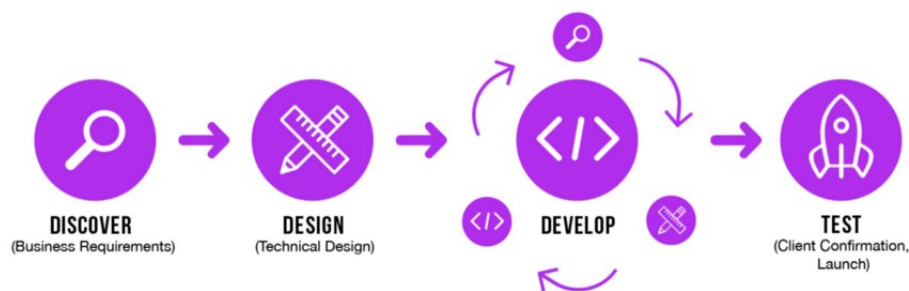


Figure 3.1: Example of Hybrid Waterfall/Agile Approach. - de los Angeles (2018)

This project was conducted using the hybrid agile-waterfall model. The researcher

Chapter 3. Project Management & Requirements

considered these methodologies alongside other alternatives such as purely agile. However, this method was selected as it gives the benefit of agile during development. Such as being able to break development into manageable chunks and produce a minimum viable product at each stage. Alongside the benefits of waterfall's organisation and systematic progress in other parts of the software development life cycle. The hybrid method outlines that planning, design and requirements gathering are completed using a waterfall model and the development, and sometimes testing, is completed using Agile sprints.

CAST (2014) also found in their study that a hybrid agile/waterfall methodology produced higher quality code compared than either method used individually.

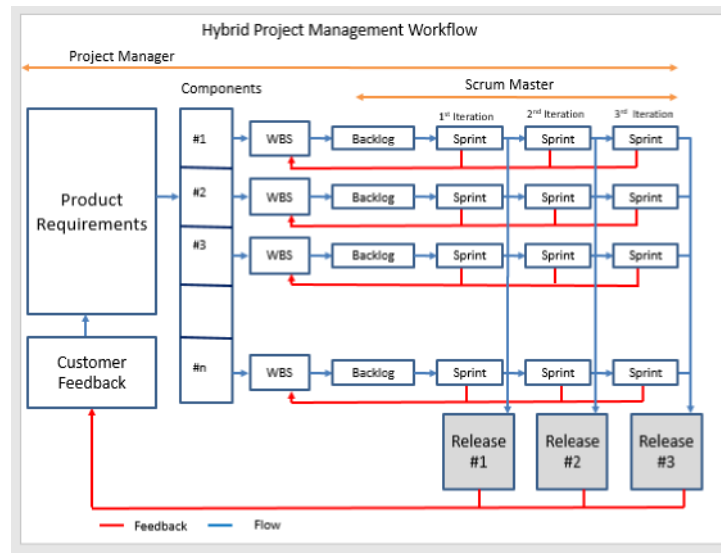


Figure 3.2: Example of Hybrid Project Management Workflow - BinFire (2018)

The structure for the workflow of this project is based on Figure 3.2. The early stages of development followed a traditional waterfall approach including the early stages requirements gathering and design, after which the development stage of the process is performed in an agile manner through the use of user stories and sprints. The testing stage is implemented after the development is complete - following the hybrid approach of water-scrum-fall.

3.2 Requirements

3.2.1 Requirements Gathering

3.2.2 Requirements from Discussions

Initial requirements gathering was conducted through unstructured interviews with the primary client, the project supervisor.

Through these interviews, as mentioned in the introduction, the target audience was identified as students studying cryptography. The inclusion of the Zodiac Killer case was discussed as a way to add narrative to the learning process. The hope was that taking a well known real-world example of a cipher that still has the mystery of being unsolved would provide an engaging storyline.

One of the objectives which were discussed was the use of visualisation in order to illustrate the cryptanalysis methods. This arose in part from findings from the literature review, where some authors found that the use of visualisations improved their results.

3.2.3 Questionnaire & Results

To gather information from the intended target audience, students who will or have studied cryptography, a questionnaire was created - the full questionnaire can be found in appendix A. The questionnaire format was chosen over an interview or another lengthy method of information gathering in the hope of achieving a higher response and participation rate. This section looks at a few of the questions which provided the most insight into possible requirements for the tool.

The goal of the questionnaire was to gain some background information on students prior experience learning cryptanalysis, their struggles, and what features they would like in a tool such as this. Seven students, all of whom had previously studied a module including cryptography answered the questionnaire.

In Figure 3.3, we can see the ciphers that the participants felt they struggled most with. Vigenère and Homophonic substitution, closely followed by Playfair. This information helps to prioritise the order the ciphers should be added to the backlog. It was

Q2 - Which ciphers did you struggle most with?

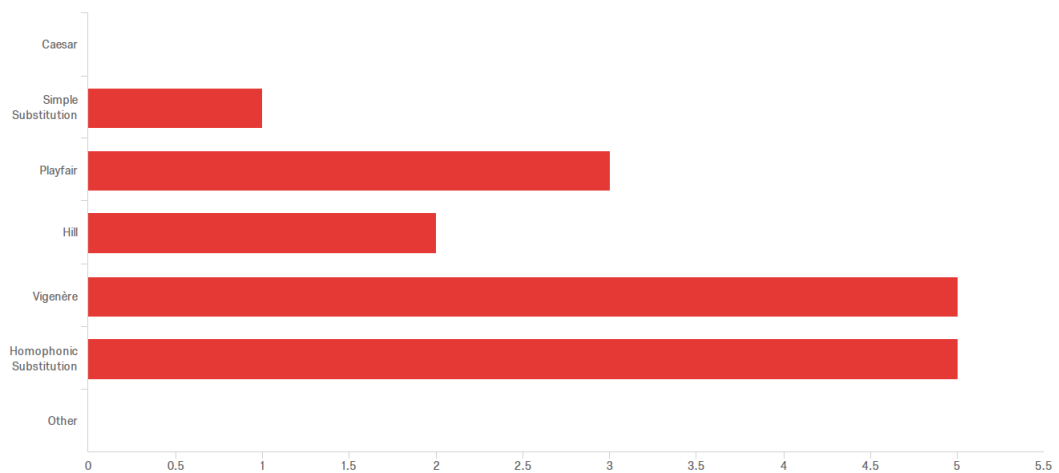


Figure 3.3: Results of *Q.2 Which ciphers did you struggle most with?*

decided that Caesar cipher and Simple substitution should be included early on as they are an excellent way to teach the basics of cryptography to beginners.

When participants were asked why they felt they struggled the responses varied including; resources being hard to follow, issues with not understanding the maths involved and difficulty spotting patterns in code.

When asked what features they would like to see, seven participants listed the following;

1. Videos which demonstrate ciphers and how they work (mentioned twice)
2. Step-by-step explanation/guides (mentioned twice)
3. Practice ciphers you can try to crack
4. Ability to select cipher type and difficulty level
5. Animations or visual explanations
6. Lots of information (presumably on ciphers)

The inclusion of videos may have been exciting, but given that there are few explanatory videos without copyright obligations, it would likely mean creating them

Chapter 3. Project Management & Requirements

from scratch. If the tool were to be adopted for teaching a class at some point the videos from the teaching materials could be integrated to provide a more varied learning experience within the tool itself. Additionally, step-by-step explanation was another repeated request which was to be included as a requirement.

Participants were also asked which aspect of these resources helped them the most, to which they answered;

1. Being able to visualise the process while being walked through step by step
2. Ability to replay parts of videos several times
3. Combination of visualisations and explanations
4. Images and information
5. Visualisation

From these responses, we can see that repetition is mentioned more than once. This draws parallels to Pellone (1995) who said "*With repetitive practice, the student will eventually perform quickly and effortlessly.*", when discussing key elements of educational software and classroom education. The ability to repetitively carry out a task or view a process until it has been learned sufficiently seems to be a key element. This is echoed by the idea of learning an idea step-by-step, as this often requires learning one aspect before the next and being able to revisit steps individually. This was applied in the design of the Zodiac Cryptogram tool in the Learn section by providing methods to encrypt and decrypt ciphers repeatedly while changing keys and shifts to experiment with the different results. In the Analysis section, it is applied by automatically updating the visualisations as the user types so they can observe the effects different analysis types have when applied to varying ciphertexts.

Additionally, the use of images, video and visualisation, in general, were mentioned - as such being able to have a visual representation of some kind to represent the processes of cryptanalysis and cryptography would be a benefit to users. 100% of participants felt that graphical visualisations or representations were useful to their learning. Additionally, 100% felt that an educational tool with visual elements explaining the process

of cryptography and cryptanalysis would be useful to their learning experience. This would be good evidence that including visual representations of cipher and analysis would be a key aspect of any software created.

3.2.4 Use Case Diagram

As part of the waterfall side of the hybrid methodology, an initial use case diagram was developed in order to better plan the functionality required. This can be seen in Appendix B. The diagram outlines the initial system design through the interactions of the user. Through this, there are a few distinct sections or paths that began to emerge. The ability to learn about ciphers, the ability to analysis ciphers via various types of cryptanalysis and the ability to analysis the Zodiac ciphers. These three paths became key pages for the Zodiac Cryptogram tool.

However, after the first iteration, the diagram was not modified further as user stories were used and updated to keep track of functionality during the development phase. As noted by Rahimian and Ramsin (2008), hybrid methodologies should follow the agile practice of prioritising requirements and tackle them in order of importance.

3.2.5 User Stories

User stories were created to illustrate the requirements needed by users of the tool. Typically they are short descriptions of the needs or wants of a user/admin/superuser etc. of the system being designed. They are usually considered part of an agile workflow. Within an agile workflow in a sprint, a selection of the user stories are added to the 'backlog' where they are moved forward until completed. Since the hybrid methodology uses an agile scrum development phase user stories are ideal for this.

As the tool has no admin, the user stories are only written from the perspective of a typical user - in this case, a student. The main functions of the tool are relatively repetitive, for example - a user should be able to perform frequency analysis on a Caesar cipher, a user should be able to perform frequency analysis on a substitution cipher, and so on. In order to represent this, a series of 'Epic' user stories were created. Epic user stories, as outlined by Cohn (2004), are large user stories that would typically not

Chapter 3. Project Management & Requirements

be feasible to complete in one iteration. The epics created for the project were;

1. A user can read about a selection of ciphers.
2. A user can encrypt/decrypt text using a selection of ciphers.
3. A user can perform a selection of cryptanalysis on a selection of ciphers

The cipher epics were broken down into smaller stories. For example in relation to a Caesar cipher the following stories were created;

1. A user can read about Caesar ciphers (history and example)
2. A user can encrypt/decrypt text using a Caesar cipher

The idea was that each cipher type would have the same functionality, and for each iteration, a different cipher would be implemented. At the beginning of the project, to ensure a functional product was created at each sprint, it was decided that each of the 'functions' that could be performed on a cipher would be completed fully for one cipher before beginning another. There was no set number of ciphers which were to be implemented. However, there was a prioritised list of the first few which were planned to be added. It was hoped that Caesar cipher, simple substitution cipher would be implemented earliest as they are a good introduction to cryptography. It is followed by homophonic substitution ciphers, as this is the type of cipher used by the Zodiac. Unfortunately, due to time, this cipher type was not implemented.

The cryptanalysis functions were broken down into;

1. A user can perform frequency analysis on a selection of ciphers
2. A user can perform Chi-squared analysis on a selection of ciphers
3. A user can perform n-gram analysis on a selection of ciphers
4. A user can perform Index of Coincidence analysis on a selection of ciphers
5. A user can perform Chi-squared analysis on a selection of ciphers

6. A user can replace text letter by letter in a selection of ciphertexts

A selection of these can be seen in Figure 3.4.

| | |
|--|---|
| <p>Story: A user can interactively preform frequency analysis on a selection of ciphers.</p> <hr/> <p>Details: This may include a bar chart or heat-map (potentially both)</p> | <p>Story: A user can interactively preform n-gram analysis on a selection of ciphers.</p> <hr/> <p>Details: This will have a series of n-gram stats such as uniqueness and repeating. These will be visualised and highlighted in the ciphertext.</p> |
|--|---|

Figure 3.4: Selection of Cryptanalysis User Stories of the Zodiac Cryptogram Tool

3.3 Project Management & Requirements Conclusion

To conclude, this project follows a hybrid methodology with a standard waterfall process until the development phase, which will be agile in nature. To aid this agile development phase user stories were used to work through the implementation of the features identified. The user stories, alongside the initial use case diagram, helped to inform the software design by identifying the needs of users and like features that could be grouped into intuitive sections.

Chapter 4

Software Design

4.1 Architecture

In this section, the structure of the Zodiac Cryptogram tool is outlined and discussed. Covering a few of the key design decisions that were made, such as the GUI framework, class structures and user interface prototyping.

4.1.1 Cross-Platform Desktop Application GUI Frameworks

One of the considerations for this project was that it be a cross-platform application. Cross-platform means the software can be run on multiple operating systems such as Windows, Linux or macOS. The benefit of this would that it by increasing the platforms the application can be used on it can increase the potential number of users.

In order to achieve this, several development approaches and GUI development options were considered. Three main paths were considered and trailed for the development of the application; Electron, Django and QT5 (Python version being PyQt5).

Electron

The Electron platform is based on Google Chromium and Node.js. It can be used to create cross-platform desktop applications using JavaScript, HTML and CCS. As it is based on Chromium, Electron operates in a similar way to a web browser. As Node.js is a JavaScript run-time environment, it has a non-blocking input/output model allowing

for multiple requests be handled at the same time, in theory allowing for the more efficient running of JavaScript code. While investigating this option, the researcher found there was little documentation or previous example of applications built using Electron as a GUI frontend and Python as a logical backend. After trying this process out, it was found to be time-consuming, and the Electron platform appeared to be taking up excessive amounts of memory when running. Additionally debugging issues within the Electron framework proved to be difficult due to the complexity of tracing errors back to their source.

Django

Django allows for the creation of web applications, which are inherently cross-platform since they run via web browsers. It is a Python web framework which is often used for rapid development. A basic application using Django as a frontend and Python as a back end was developed in order to assess the feasibility of its implementation within the project. It was concluded that although it was very much possible to create an application such as this within the Django framework. However, the time needed for the researcher to advance their JavaScript knowledge to an appropriate level to create the visualisations for the cryptanalysis would be outside the timescale available.

Qt5

Qt5 is a cross-platform application framework which is written in C++. Its widget toolkit and designer tool are often used for the development of Graphical User Interfaces (GUI). This toolkit allows for rapid development of UIs which have a native look and feel. After trailing PyQt5, it was found to be most suitable for the project.

The decision to use PyQt5 to create the UI was based on its ease of use, a large amount of documentation and online support, as well as it not requiring the researcher to learn new languages to use.

4.1.2 Qt5 Framework

4.1.2.1 Cross-platform

One of the main benefits of the Qt Framework is its ability to mimic the native appearance of the platforms it runs on. Executables for Windows, Linux and MacOS can be created using the framework.

4.1.2.2 User Interfaces

Within the Qt Framework, there are three main technologies for building user interfaces; Qt Quick, Qt Widgets and Qt WebEngine. As the creation of a web application had been decided against the comparison between Qt Quick and Qt Widgets was considered. Qt Quick creates interfaces using QML (Qt Modeling Language), which is an object description language. Qt Widgets is a module consisting of a comprehensive collection of UI components. These components are styled to create a native desktop feel. The widgets have been in use longer than Qt Quick, the widgets have more documentation and were thus decided to be the more suitable option.

Abstraction of GUI Qt uses native style APIs for the platform it is being run on to achieve its native appearance on multiple platforms. Qt5 also comes with a 'designer' which allows you to drag and drop elements onto the UI and generate UI files which are written in XML (eXtensible Markup Language) and hold information on the GUI contents and layout. These UI files can be loaded and referenced within python files, along with the elements such as buttons within them.

Signals and slots

Qt provides a functionality known as slots and signals. It essentially works as a notification system between objects. For example, if a user changes a slider's position it emits a signal, we might assign a slot to another object - like a text field - to receive this signal and in turn respond in some way - such as displaying the slider value. Qt comes with a number of predefined slots and signals already built into its Widgets.

However, sub-classing Widgets can allow us to assign custom signals and slots Qt5. Additionally, one signal can be received by more than one slot.

PyQt5

PyQt5 is a module with a collection of Python bindings for the Qt GUI framework which allows Python to be used for developing in Qt rather than C++. As the researcher has experience with the Python programming language, using the PyQt5 bindings was the natural choice in order to use Qt in this project.

4.1.3 Model-View-Controller

The Model-View-Controller (MVC) design pattern was used in the architecture of the Zodiac Cryptogram tool. MVC is a relatively loosely defined pattern and is also referred to as layered architecture.

As described by Deacon (2009) the MVC structure is as follows;

- **Model** The model should be an isolated, unchanging section of the application with classes which model and support the core purpose of the application.
- **View** The view is often, but not always, the GUI and displays information from the model to the user. It is often said to handle the output of the model.
- **Controller** The controller conversely handles the input from the user and manipulates the view via the model.

The benefits of using a design pattern such as this are that it helps to encourage class design which has more defined and individualised responsibilities — ultimately leading to classes which have less code repetition, more reusability and more flexibility later in development. Additionally, it makes maintenance later in the Software Development Life-cycle easier as the added separation makes debugging, testing and updating more manageable.

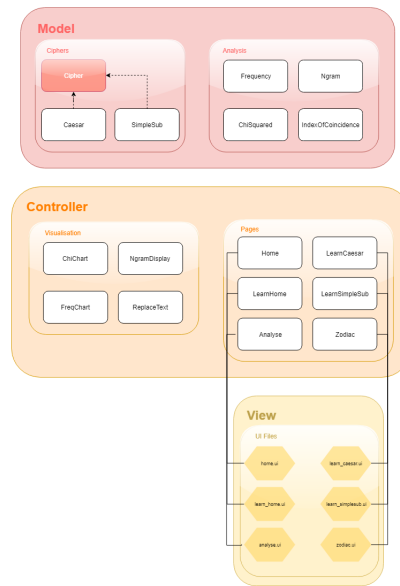


Figure 4.1: High-level view of MVC design of Zodiac Cryptogram

Figure 4.1 illustrates how the architecture of the tool has been created using the MVC pattern. First, the model, which is where the primary logical operations are preformed, contains the *Cipher* classes and the classes which fall under the banner of *Analysis*.

The controller, which handles input, holds the *Pages* and *Visualisation* classes. Each of the *Pages* classes are responsible for loading their individual *UI Files* and handling the various inputs and passing them to the model or view. The *Visualisation* classes each handles the manipulation of data from one of the *Analysis* types before passing it to the relevant view.

Qt Designer was used to design and generate UI files which act as the view. Qt Designer is a tool made by Qt for creating GUI files using the Widgets provided by the Qt framework. As mentioned briefly before, the Widgets within these UI files can be referenced and accessed in python files (which will act as part of the controller).



Figure 4.2: High-level view of MVC design of Zodiac Cryptogram

4.2 Class Design

The following section details the classes created within the design of the tool. An object-oriented approach was used for architecture design. The goal was to minimise code re-use and create an easier to maintain the final product.

The model-view-controller diagram in Figure 4.1 outlined the intended class structure. Each of the titles; Ciphers, Analysis, Pages and Visualisation within the diagram became individual modules used in the tool, as shown in Figure 4.2.

4.3 User Interface Design

Braude and Bernstein (2016) consider User Interface Design to be a part of the requirements gathering process, although others often include it with the 'Design' phase of a project. Within this project, prototyping was used both to inform and confirm the inclusion of functionality and to plan how the logical processes would be displayed - thus informing class design on a more granular level such as return types. Hence, it is considered to technically be part of both despite it being within the Software Design chapter.

The design of the User Interface is a vital part of any learning software. If the interface is hard to understand or has a steep learning curve, it will negatively affect a users ability to learn from the tool. Taking into account the earlier objectives, it was established that visualisation of the ciphers, analysis and their process would be useful to learning. As Schweitzer and Baird (2006) said pedagogical visualisations of abstract concepts has proven to be effective in multiple disciplines.

4.3.1 User Interface Design Principles and Choices

Several of the UI design choices were informed by information gained from the literature review. Firstly, the division of the tool into the 'Learn' and 'Analyse' sections was drawn from the similar structures used in Crypto-Tutor (Luburi et al. (2016)), VigVisual (Li et al. (2015)) and the work of Yang et al. (2009) on Theory-Algorithm-Practice-Application learning flow.

4.3.2 Prototyping

In order to better anticipate user interaction and assess if the software would fulfil the initial objectives, a basic paper prototype was created.

After sketching the user interface on paper, taking into account the previously outlined requirements, the UI design was laid out in AdobeXD. AdobeXD is a free UI designing and prototyping software which allows for rapid creation of graphical user interface layouts. Two designs were created within this tool. The first Prototype 1,

Chapter 4. Software Design

which can be seen in Figure 4.3 was designed with a look and feel similar to that of a browser-based application.

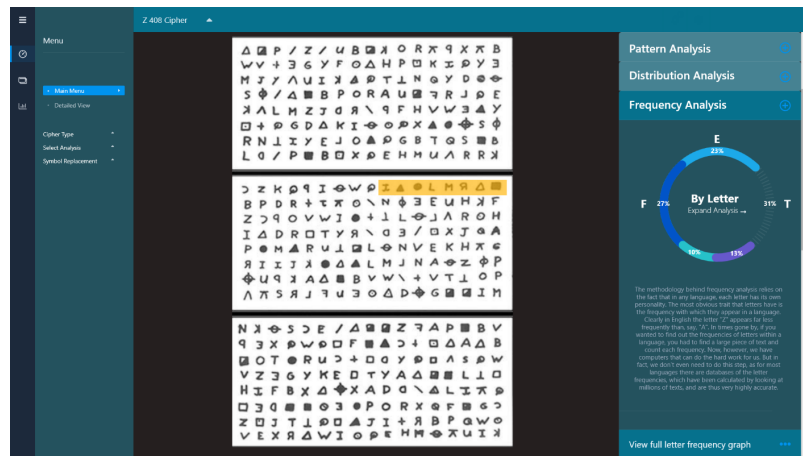


Figure 4.3: Prototype 1 User Interface for Zodiac section of tool created in AdobeXD

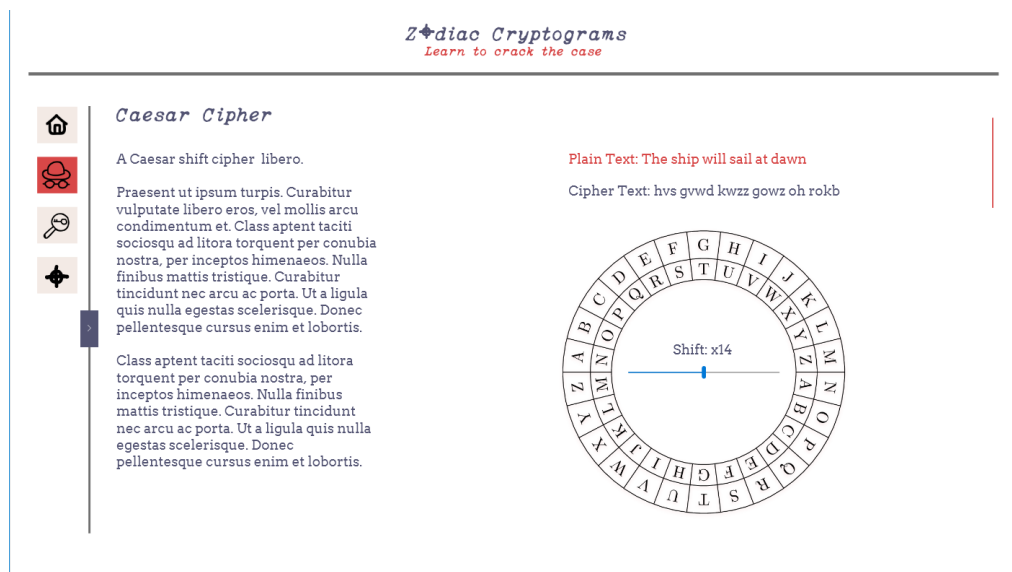


Figure 4.4: Prototype 2 User Interface for Learn Caesar page created in AdobeXD

Figure 4.4 shows the 'Learn Caesar' page of Prototype 2. Neither prototype was a completed MVP as initially it was decided the number of features, cipher types and analysis types would depend on the progress of the development iterations within the given timescale. As such, the prototypes outlined at least one cipher in the learn section and a series of analysis types, with their potential visualisation in the analysis

section. Through the prototype, basic layouts which could be consistently repeated were experimented with. In designing the prototypes, it was decided sidebar navigation showing all the main pages was a good way to provide quick and clear navigation to users. The main sections that would be needed to fulfil the initial user stories were also identified as Home - to provide information about the tool and its use, Learn - for the user to learn about individual ciphers, Analyse - to analysis a variety of ciphers, and Zodiac - to hopefully learn about and analyse some of the Zodiac ciphers.

4.4 Software Design Conclusion

In summary, the architecture of the tool is modelled following the MVC design pattern. This reinforced by the used of the PyQt5 Framework which is used to generate the UI files in XML. The class design was split into the modules *Analysis*, *Ciphers*, *Pages*, and *Visualisations*, with the ui files in another module. Through the prototyping process of the GUI, the sections of the tool and general visual layout was decided.

Chapter 5

Implementation

5.1 Implementation Introduction

The implementation section will discuss the development of the Zodiac Cryptogram tool in terms of coding choices and the methods through which the structure outlined in the software design section was executed. The construction of the various cryptanalysis features will be explored as well as discussing libraries used within the tool.

5.1.1 Programming Language & Coding Standard

Python 3.7 was used to develop the Zodiac Cryptogram tool. At the time of beginning development, Python 3.7 was the most recent release available.

To maintain a consistent and clear style throughout the application, the PEP8 (Python Enhancement Proposal) style guide for Python was selected as a coding standard to adhere to. In order to help maintain this standard throughout the programming process, the Pylint plugin for Pycharm was used.

5.1.2 JetBrains Pycharm IDE

The PyCharm Professional Edition IDE was used to develop the tool. It was selected in part because it is a widely used open-source platform for developing python projects. It also features an integrated python debugger and unit-testing capabilities which make the development process more streamlined. Version control is an essential aspect of

Chapter 5. Implementation

software development. The Pycharm IDE allows for easy integration with various Version control systems (VCS). In this project, Git was used to track files and changes along with a GitHub repository which allowed for the backing up of files in an online location for additional security.

5.1.3 Library Use

In the development of the Zodiac Cryptogram program, two libraries - outside of the Python standard libraries and PyQt5 as discussed - were used to assist in creating some of the visualisations used for the software.

5.1.3.1 NumPy

Numpy provides a range of options for performing scientific computing in Python. Within this project, its array options were often used in order to prepare data generated by the 'Cipher' and 'Analysis' functions before being visualised. As the main method of visualisation, Matplotlib utilises NumPy arrays using NumPy to sort data made it easier to pass data to the Matplotlib functions.

5.1.3.2 Matplotlib

Matplotlib v3.1.0 is a Python library for plotting 2D figures. Within the matplotlib API is a backend for Qt5 (`matplotlib.backends.backend_qt5agg`), which enables the use of embedded plotting in the Qt5 GUIs. The ability to embed the visuals was essential to creating a more seamless user experience.

The researcher experimented with two additional options for visualising the graphs within the tool. Firstly, Qt Chart and Qt Data Visualization were explored. These are both add-ons within the Qt Framework. Qt Chart proved difficult to work with and had limited options for altering the appearance of graphs. As such, it was quickly ruled out for use in the project. Secondly, an external library called PyQtgraph, which was built on PyQt4 and NumPy pyq (2019). It was designed to provide faster plotting and realtime data display. In observation, this did prove to be true in comparison to the

same data being plotted by Matplotlib. However, matplotlib offered more flexibility in its implementation and better documentation, which made development faster overall.

5.1.4 Analysis

In the project, it was hoped that as many analysis types would be added as the time scale allowed for. Consequently, the following analysis classes were created; Frequency, Ngram, ChiSquared, and IndexOfCoincidence.

5.1.4.1 Frequency

The frequency class is used to calculate the frequency of words or letters in a given piece of text. This was implemented using a simple Counter. The options to allow for removal of spaces and punctuation within the text and to option to switch between letter and word frequency. This particular functionality was not controllable by the user in the final product due to time constraints but could be implemented at a later date.

As part of this class, it was required that it was possible to compare an observed frequency with an expected frequency of the English language. Expected frequencies can be calculated for any language by counting the letters that occur within a large sample of text in that language working out the percentage of the time each letter appears in said text. However, the letter frequency calculated will be marginally different depending on the characteristics of the text used to generate the letter frequencies and the exact method used to calculate the frequencies. For the English language, one of the most commonly used was calculated by Mayzner and Tresselt (1965) using a corpus of 20,000 words from a range of sources. However, the researcher chose to use a set of letter frequencies generated by Norvig (2013) - who was asked by Mayzner to expand on his work from 1965 - using Google books data as the reference corpus which contained 743,842,922,321 words. These frequencies can be seen in Figure 5.1. The choice to use these frequencies was due to firstly, the data set being much more extensive, giving a more comprehensive coverage of the English language. Secondly, since the data set included books written up to 2013 this would mean text written in a more modern style

Chapter 5. Implementation

would be included. As the characteristics of language change over time, so would the typical frequency of letters within that language.

Within the next few iterations being able to implement the ability to hover over the bars on the observed chart and see the values displayed, possibly also highlighting the few values closest to them on the expected chart would be an objective that will make the chart quicker to use.

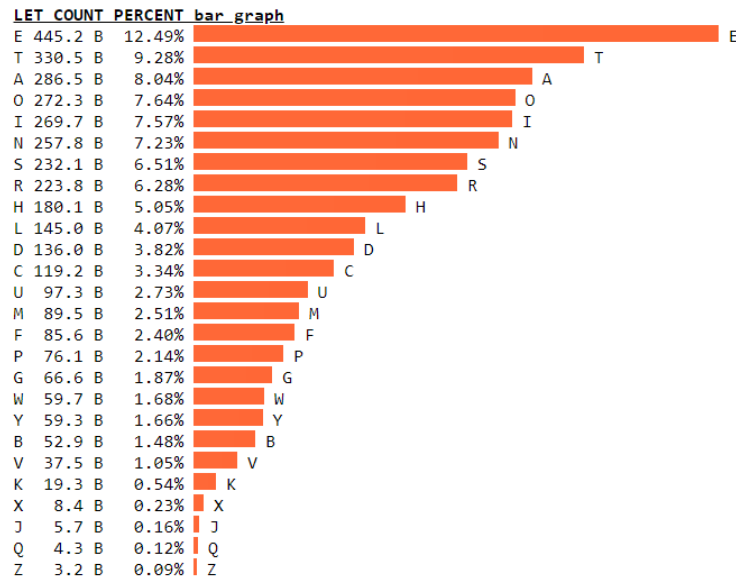


Figure 5.1: Letter Frequencies based on data from Google Books Norvig (2013)

5.1.4.2 Ngram

The Ngram class works by generating Ngrams from a given piece of text. The class was designed to generate ngrams of any given n. However, only options of up to n=4 (quadgrams) were created as easy to access get methods. The ngrams can also be generated as word ngrams - which look for repeating words, or letter ngrams - the ngrams looking for repeating letter groups. This is because it is unlikely users would find the need to go over this level as ngrams higher than this are more uncommon. The class has options to retain or remove spaces and punctuation individually from the text, again as with frequency, these options were not made available to the user in the current version.

The ngrams are generated by splitting the given text either by words or letters depending on mode. A list of characters is created, and then each character is added to an array as a tuple with the n number of characters after it, i.e. for bigrams [a,b,c] would produce [(a, b),(b, c)] and for trigrams it would produce [(a, b, c)].

A slight issue with the `get_frequency_ngrams()` method, it is intended that it would display the count of the times the given n-gram appears in the text. As it uses the `most_common()` method from Counter, it will return as expected when used with mono-grams, but the number is inflated with other ngram types if several of the same characters are used in a row. i.e. 'aaaa' will return a count of bigrams 3 rather than 2, as it is calculated as [(a, a), (a, a), (a, a)], while technically true this can be confusing to those trying to decrypt a ciphertext. This function was added later in development, and as such, there was not enough time to thoroughly test and solve the issue. The source files for common English ngrams were source from practicalcryptography.com (2019).

5.1.4.3 Chi Squared

Chi-Squared, as mentioned in the literature review, is a method of comparing two probability distributions. In this case, we are comparing the expected frequency of letters for the length of inputted text against the actual 'observed' frequency of letters we see within that text.

$$\chi^2 = \sum_{i=A}^{i=Z} \frac{(O_i - E_i)^2}{E_i} \quad (5.1)$$

Equation 5.1, from Barter (2017a) shows the Chi-Squared Formula. Within the cipher class the Chi-Squared formula was broken down into sections and a method created to calculate each stage. This was done so that if any stage of the process was to be visualised or examined, it would be easy to do so. The main functions in the class return;

1. O_i The observed letter count
2. E_i The expected letter count

3. $\frac{(O_i - E_i)^2}{E_i}$ The difference between the observed and expected squared, and divided by the expected - for each letter
4. $\chi^2 = \sum_{i=A}^{i=Z} \frac{(O_i - E_i)^2}{E_i}$ The sum of the previous

Typically cryptanalysts would concern themselves mainly with the overall sum, shown in point 4, but having the option to compare the counts letter by letter can both illustrate what one would expect to see distribution wise within an English substitution cipher and can also identify letters which have unusually high distributions.

5.1.4.4 Index of Coincidence

$$I.o.C = \frac{\sum_{i=A}^{i=Z} C_i(C_i - 1)}{L(L - 1)} \quad (5.2)$$

The index of coincidence class performs the following methods to calculate the I.o.C of a given piece of text, based on Equation 5.2 from Barter (2017b);

1. C_i The observed letter count
2. $C_i(C_i - 1)$ The observed letter count multiplied by itself minus one
3. $\frac{\sum_{i=A}^{i=Z} C_i(C_i - 1)}{L(L - 1)}$ The sum of the previous divided by the length of the alphabet used L multiplied by itself minus one

There is also a function which will return a string of the most likely language based on the current index of coincidence of the text entered. To calculate this, the I.o.C. is compared against the values for a selection of languages, shown in Figure 2.3 in the Literature Review.

5.1.5 Ciphers

An Abstract Base Class (ABC) was used to create a 'cipher' class for all cipher types to inherit from. ABCs can be used to create an interface type class for custom objects in python. This structure was used to create a consistent implementation of methods throughout all the cipher classes. It is intended that the creation of future ciphers

Chapter 5. Implementation

```
import string
import abc

class Cipher(abc.ABC):
    """
    Base class for all ciphers
    """

    @abc.abstractmethod
    def encrypt(self, t_string):
        """
        Takes a variable and returns an encrypted string
        """
        return string

    @abc.abstractmethod
    def decrypt(self, t_string):
        """
        Takes a variable and returns a decrypted string
        """
        return string

    @abc.abstractmethod
    def upper_az(self, t_string, punctuation_off=True, space_off=False):
        """
        Takes a variable and returns a list of characters in uppercase.
        Punctuation and spacing can be turned off or on.
        """
        return_val = str(t_string)
        if punctuation_off:
            # removes punctuation, keeps space
            return_val = return_val.translate(str.maketrans('', '', string.punctuation))
        if space_off:
            # removes spaces
            return_val = return_val.replace(' ', '')
        # returns uppercase of string with only letters and spaces - special chars removed
        return ''.join(letter for letter in return_val.upper()
                        if letter.isalpha() or letter.isspace())
```

(a) Code of 'Cipher' ABC class

```
from ciphers import cipher
import string
import random

class SimpleSubCipher(cipher.Cipher):

    def __init__(self):
        super().__init__()
        self.alphabet = string.ascii_uppercase
        self.key = self.gen_key()

    def set_key(self, key):
        assert len(key) == 26 and key.isalpha()
        self.key = key

    def get_key(self):
        return self.key

    def gen_key(self):
        key = ''.join(random.sample(self.alphabet, 26))
        self.set_key(key)
        return key

    def encrypt(self, t_string):
        t_string = self.upper_az(t_string)
        cipher_dict = dict(zip(self.alphabet, self.get_key()))
        result = ''.join(' ' if letter == ' '
                          else cipher_dict[letter]
                          for letter in t_string)
        return result

    def decrypt(self, t_string):
        t_string = self.upper_az(t_string)
        cipher_dict = dict(zip(self.get_key(), self.alphabet))
        result = ''.join(' ' if letter == ' '
                          else cipher_dict[letter]
                          for letter in t_string)
        return result

    def upper_az(self, t_string, punctuation_off=True, space_off=False):
        return super().upper_az(t_string, punctuation_off, space_off)
```

(b) Code showing 'SimpleSubCipher' using 'Cipher' ABC

would always use this base class so that there is a reliable expectation that there will be an 'encrypt()' and 'decrypt()' function and they will return a string when called.

As you can see in Figure 5.2a the abstract methods have been implemented in the 'SimpleSubCipher' class, as well as the 'upper_az()' method is inherited from the 'Cipher' superclass.

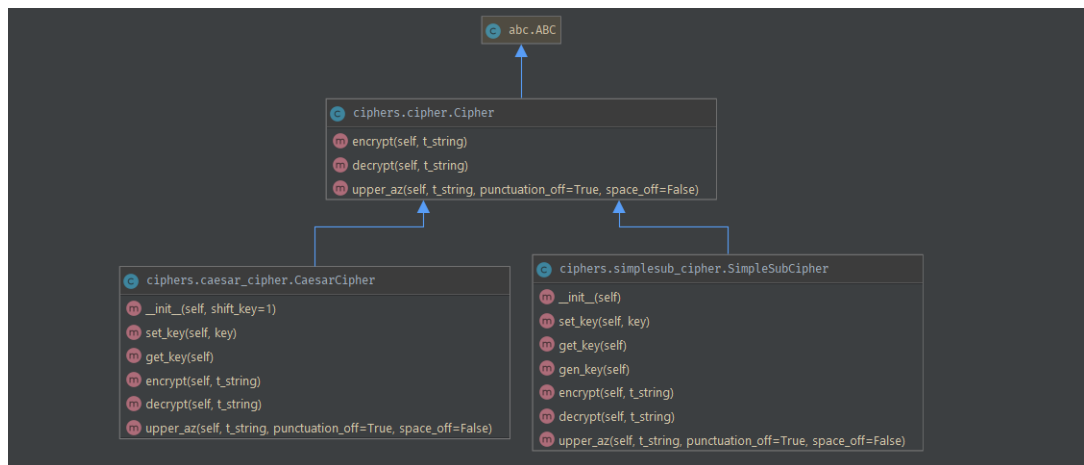


Figure 5.3: Class Structure for Ciphers

5.1.6 Pages

The Main UI was created as a large, main window which holds the title of the tool and the toolbar down the left-hand side of the window. PyQt5 only allows one main window to be run at a time. There are several ways in which you can create multi-page applications within Qt. The researcher opted to use a `QStackedWidget`, which can hold multiple widgets that can act as pages. In order to assign a class to one of these page Widgets, the widget must first be "promoted" inside Qt Designer. Promoting a widget is a way of creating a custom widget which uses an existing widget type as a base. Classes can then be made of these promoted Widgets in order to customise them when additional functionality is required.

In the case of the pages, `QWidget` was used as the base. Figures 5.4 shows how the main UI, with the Learn Caesar page nested within a `QStackedWidget` appears at runtime. A side effect of this method is that keeping track of the current page requires a variable within the main window class. Potentially this could be done via the signals and slots provided by PyQt5, but when implementing this, the researcher found it required more code in all of the other Page classes as opposed to a single function within the main window class. Hence, the decision to implement the page tracking solely via the main window class was made.

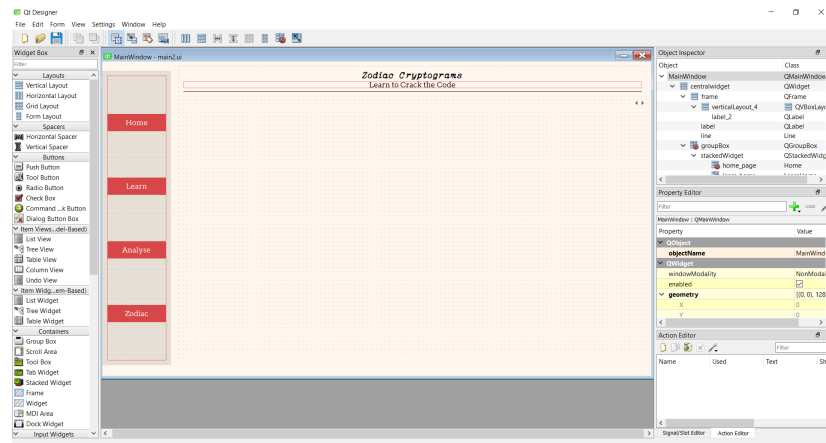
Chapter 5. Implementation

The Analyse page became responsible for controlling a substantial part of the Zodiac Cryptogram logic as it loads all the charts within the analysis tool QTabWidget. At one point during development, it was noticed that when changing between tabwidget pages, there was a time-lapse between the loading of the tabs. After debugging using the Pycharm inbuilt debugger it was determined this was due to an overburdened method which handled the logic in the tabs. After altering the method to reload only the logic and visuals on the currently open tab, the lapse was almost entirely removed.

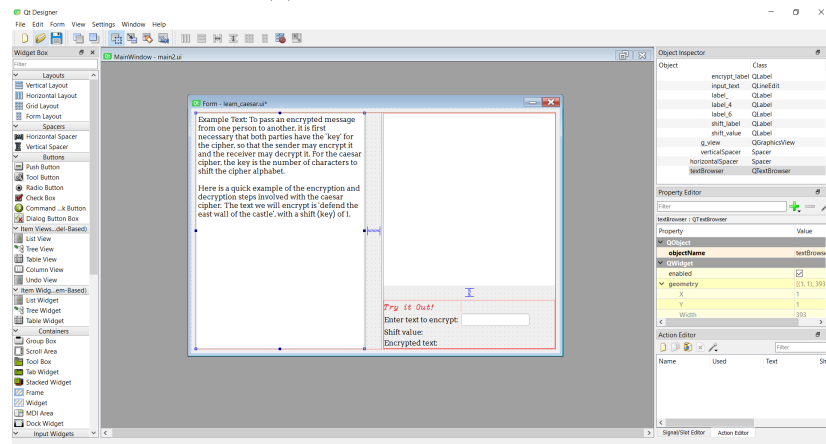
Originally, it was planned that each cipher type would have its own analysis page, featuring only the types of analysis which would be most relevant to that cipher. After the first couple of sprints, it became clear that it would be more efficient to have all the analysis functionality in one place. This was because often the analysis types were effective in some way for multiple ciphers and being able to compare the results of one type of analysis on several cipher types was useful for learning about the cryptanalysis processes. The overarching user stories for this stayed the same with 'a user can perform frequency analysis on a variety of cipher types' still being applicable, but the more refined 'a user can perform frequency analysis on a caesar cipher' being dropped. While this was still technically something that could be done, it was too detailed to be useful after the design change.

The text within the pages is relatively basic and kept simple so as to be easily understood. In future the text could be expanded upon to include more detailed descriptions and be honed for its purpose further. Within the Analysis page the text is loaded from HTML files which change depending on the current open tab. The Zodiac page is lacking in the history behind the Zodiac currently and text about the cipher has been left empty. It is hoped this section of the text can be built upon in future iterations.

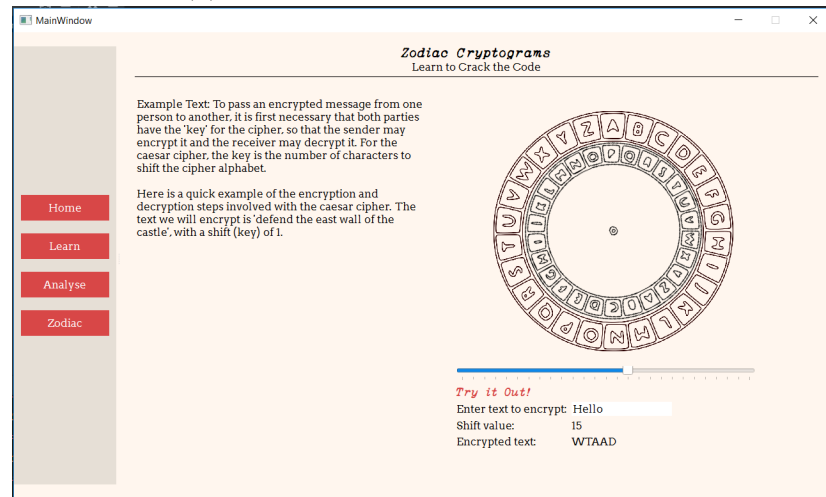
Chapter 5. Implementation



(a) Main UI in Qt Designer



(b) Learn Caesar page UI in Qt Designer



(c) Combined result of Main UI and Learn Caesar UI when run as program

Figure 5.4: Multiple UI files in Qt Designer combined to create UI at runtime

Chapter 5. Implementation

```
import os
from PyQt5 import QtWidgets, uic
from PyQt5.QtCore import pyqtSlot, pyqtSignal, Qt
from functools import partial
from ciphers import simplesub_cipher

root = os.path.dirname(os.path.abspath(__file__))
UIFile, Path = uic.loadUiType(os.path.join(root, "../ui/learn_simplesub.ui"))

class LearnSimpleSub(UIFile, Path):
    selected_page = pyqtSignal(int)

    def __init__(self, parent=None):
        super(self.__class__, self).__init__(parent)
        self.setupUi(self)
        self.sub_cipher = simplesub_cipher.SimpleSubCipher()
        self.init_ui()

    def init_ui(self):
        table = self.text_table
        for i in range(table.columnCount()):
            table.horizontalHeader().setResizeMode(i, QtWidgets.QHeaderView.ResizeToContents)
            table.horizontalHeader().setResizeMode(i, QtWidgets.QHeaderView.Stretch)
        self.encrypt_button.clicked.connect(partial(self.encrypt))
        self.decrypt_button.clicked.connect(partial(self.decrypt))
        self.gen_button.clicked.connect(partial(self.gen_key))
        self.gen_key()

    @pyqtSlot()
    def gen_key(self):
        key = list(self.sub_cipher.gen_key())
        for i in range(self.text_table.columnCount()):
            self.text_table.setItem(0, i, QtWidgets.QTableWidgetItem(key[i]))
            item = self.text_table.item(0, i)
            item.setFlags(item.Flags() ^ Qt.ItemIsEditable)

    @pyqtSlot()
    def encrypt(self):
        cipher_text = self.sub_cipher.encrypt(self.input_text.text())
        self.output_text.setText(cipher_text)

    @pyqtSlot()
    def decrypt(self):
        cipher_text = self.sub_cipher.decrypt(self.output_text.text())
        self.input_text.setText(cipher_text)
```

Figure 5.5: Code of LearnSimpleSub Page

5.1.7 Visualisation Implementation

The majority of the graphic visualisation was implemented using Matplotlib. Some such as the Cipher Disk were implemented entirely using the PyQt5 Widgets.

5.1.7.1 Cipher Disk

On the *Learn Caesar* page a spinning cipher disk, as seen in Figure 5.6 was imple-

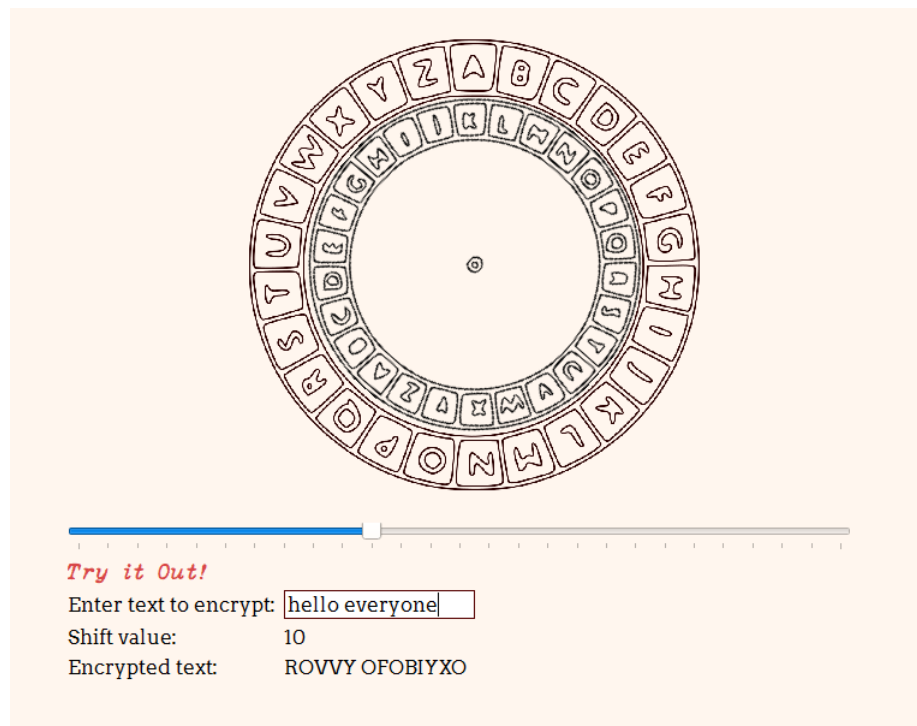


Figure 5.6: Rotating cipher disk as shown on the Learn Caesar page

mented to demonstrate the simple theory behind the Caesar shift. The cipher disks are two png images with transparent backgrounds. One is of the inner disk and the other the outer. Initially, this was implemented by creating two QPixmaps to hold the images and overlaying them. The outer pixmap was animated to rotate using basic transform property. It had been hoped that by changing the settings of the overlaid Q pixmap to show the inner display through the transparency would be possible. However, after further research, it was uncovered that within QT5 when overlaying images, it is recommended to use a QGraphicsView. Within a QGraphicsView multiple images can be added and interact with each other. After trying several methods of adding the animation - such as property animation - it was found that using a QTimer to stop and start the animation was the best method. In order to rotate the image to the right interval the rotation is set to $-\text{shift value} \times (360 \div 26)$.

The rotation animation is controlled by the user changing the value of a slider. The Slider class used was built upon the QSlider class from PyQt5 in order to modify the signal emitted by the slider when its value changes. The custom slider will emit a signal

of its value upon changing - which can be caught by a slot in the Learn Caesar page class allowing it to use this value to update its display label and pass to the Caesar cipher class in order to use it to calculate the encryption of text enter by the user.

5.1.7.2 Frequency Chart

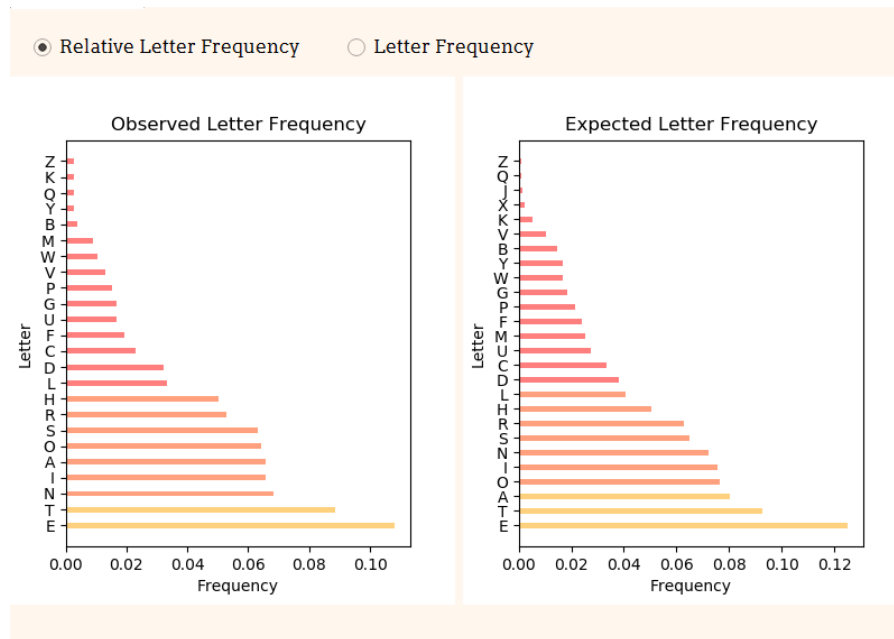


Figure 5.7: Example of Frequency Visualisation using section of a speech by President Jimmy Carter

Initially, it was difficult to correctly scale the graph and fit the letters used within the space. Adjusting the width of the bars along with setting the axis to auto-scale helped with this. Additionally assigning the letters as 'ticks' along the y-axis proved to be challenging as positioning them next to the correct data proved difficult. This was solved by using the `set_y_ticklabels()` using the data dictionary keys, which were letters and the values the plotted data, which aligned the ticks and the data correctly. The code relating to the plotting of the chart can be seen in 5.8.

A side by side comparison between expected relative letter frequency in the English language and the observed relative letter frequency from a user inputted text (or ciphertext) was used to allow for easy comparison. An example of this in action is shown

Chapter 5. Implementation

in Figure 5.7. Three colours red, orange-red and orange were used to add visual distinction to value ranges within the graph. While the value ranges are relatively arbitrary, <0.4 , $0.4 - <0.8$, and ≥ 0.8 , by having some distinction within the graph it is hoped it will make it easier to focus on similar values. This can be viewed in Figure 5.8. There is also the option to have the chart display letter frequency as counts of letters only. The *FreqChart* class is used to plot both the observed and the expected letter frequencies. To plot expected English values a boolean value can be assigned upon initialising the class.

```
def plot(self):
    """Plots graph with 3 coloured value ranges"""
    data = self.pd
    self.axes.cla()
    plot_ax = self.figure.add_subplot(111)
    if self.expected_freq:
        plot_ax.set_title('Expected Letter Frequency')
    else:
        plot_ax.set_title('Observed Letter Frequency')
    plot_ax.set_ylabel('Letter')
    plot_ax.set_xlabel('Frequency')
    x = np.array(range(len(data)))
    y = np.array(list(data.values()))
    color_val1 = (y < 0.04)
    color_val2 = (y >= 0.04) & (y < 0.08)
    color_val3 = (y >= 0.08)
    plot_ax.barh(x[color_val1], y[color_val1], 0.35, align='center', alpha=0.5, color='red')
    plot_ax.barh(x[color_val2], y[color_val2], 0.35, align='center', alpha=0.5, color='orangered')
    plot_ax.barh(x[color_val3], y[color_val3], 0.35, align='center', alpha=0.5, color='orange')
    plot_ax.set_yticks(x)
    plot_ax.set_yticklabels(list(data.keys()))
    plot_ax.width = 1
    plot_ax.autoscale(True)
    self.draw()
```

Figure 5.8: Code snippet of `plot()` from 'FreqChart' which sets up and plots the chart

5.1.7.3 Ngram Display

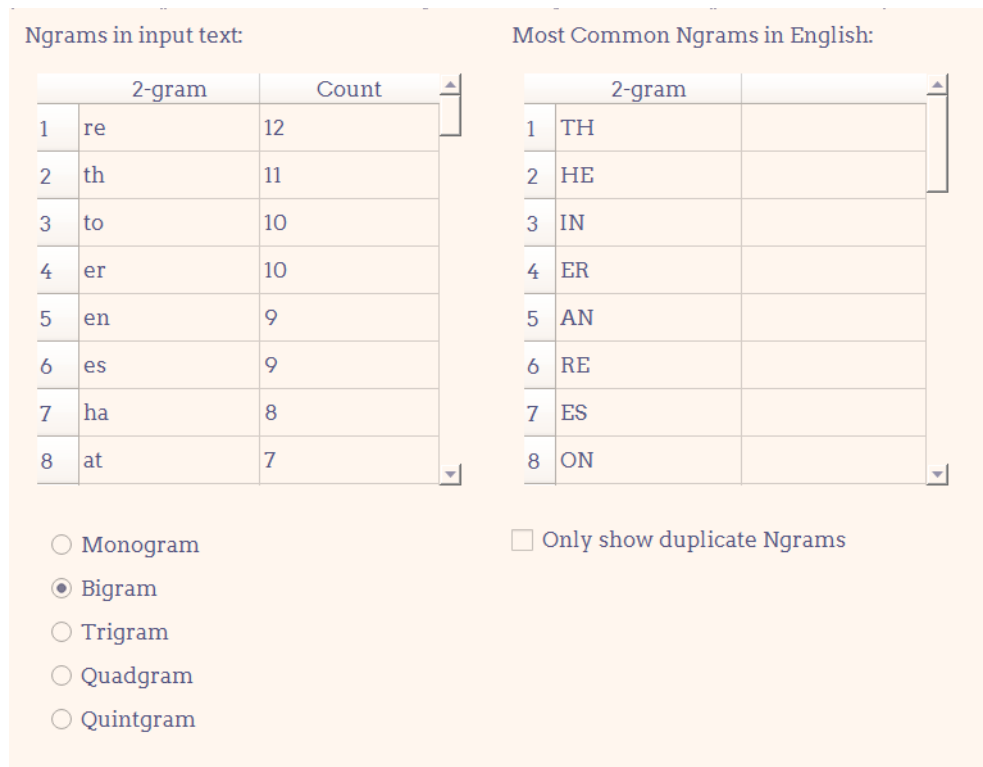


Figure 5.9: Ngrams displayed to help work out possible replacements

The Ngram display consists of two scrollable `QTableWidgets`. The first of which is used to display the Ngrams found within the text inputted by a user. The second is used to display up to the thirty most common Ngrams in the English language for the current type of Ngram selected. Radio buttons allow the user to select the type of Ngram they wish to display. The method of generating Ngrams used means that all combinations of letters are returned *i.e.* *'undercover'* would give you *'un'*, *'nd'*, *'de'*, *'er'*, *'rc'*, *'co'*, *'ov'*, *'ve'*, *'er'* when looking at Bigrams, whereas a user may just want to focus on repeating Ngrams. As such, there is a checkbox to show only the reoccurring Ngrams, which in the case of *'undercover'* would return only *'er'*.

The Ngram tool displays the most common English N-grams, to source these files were used from practicalcryptography.com. Initially, when trying to load these Ngrams the tool ran into issues with the size of the files when loading. Additionally, it was rather overwhelming to display even half the list of common Ngrams, as such it was decided that only the first 30 (most common) Ngrams would be displayed - which meant

Chapter 5. Implementation

the code only had to iterate over the first 30 lines rather than the full sets.

If time had allowed the addition of more colour to help, users identify possible replacements or to have the tool suggest the most likely replacements for the first three Ngrams would have been helpful.

5.1.7.4 Chi Chart

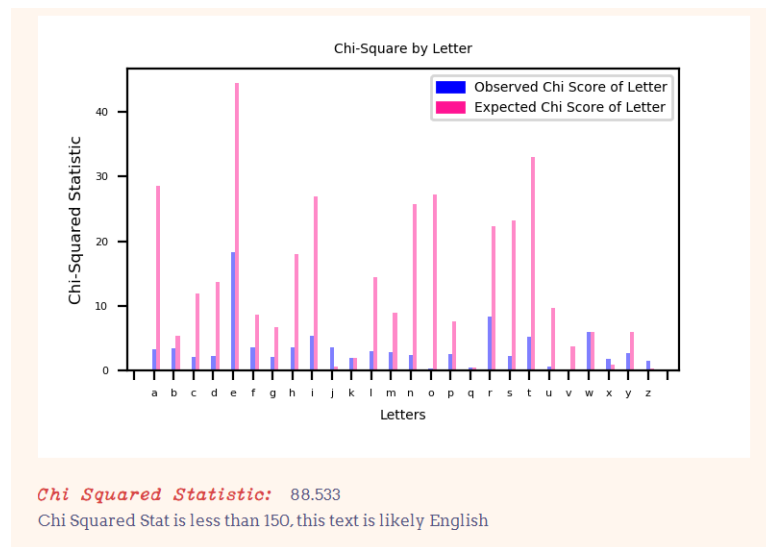


Figure 5.10: Code snippet of `plot()` from 'FreqChart' which sets up and plots the chart

Chi Statistics are not commonly visualised as individual letter values. Usually, the final chi statistic is the main focus. The researcher chose to visualise the individual letter scores against the expected individual letter scores to illustrate better the process of getting to the final summed score. Additionally, this method makes it easier to spot letters which are occurring at a level that is notably higher, for example, if an 'X' is accounting for the highest statistic it is likely this would map to a letter with a much higher English occurrence such as 'E' or 'T'. This is of course, not a replacement for frequency analysis, but using the two in conjunction can be useful. In order to plot two sets of data on the one graph, the axis had to be updated with each new value and the bar offset by half the width of the original plotted bar to place them side by side. As mentioned previously if the overall statistic is 150 or below, then it is generally

Chapter 5. Implementation

considered that this text matches the expected structure of English. This is displayed via a QLabel to show if the text is likely to be English or not.

5.1.7.5 Index of Coincidence Chart

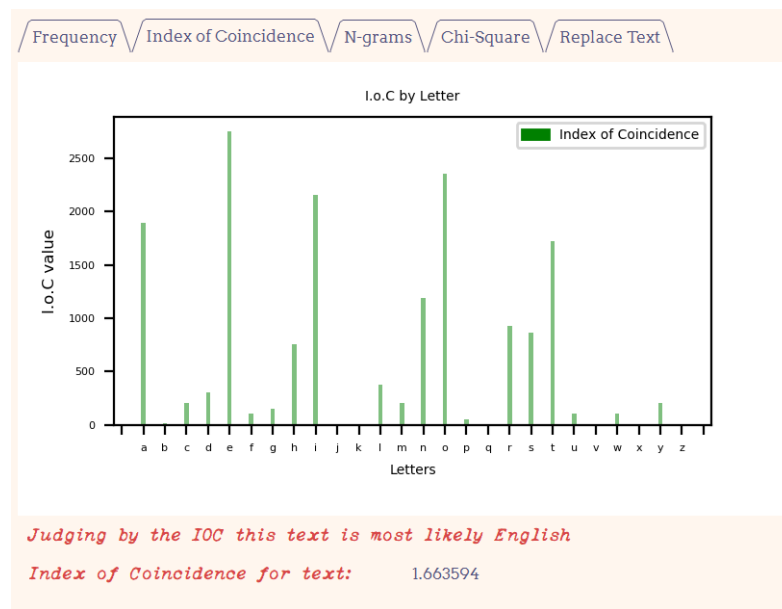


Figure 5.11: Index of Coincidence chart showing the I.C. for Martin Luther King's 'I have a dream' speech

The index of coincidence chart was implemented in much the same way as the chi-squared chart, only with one set of data rather than two. Potentially the code could have been simplified and re-written to allow for reuse of the chart for both chi-squared and index of coincidence, but there was not enough time to fully explore this. At present, the tool will update a QLabel to display the language the text provided is most likely to be written in, based on the normalised I.o.C. calculated. This is limited to the values outlined in Figure 2.3 in the Literature Review.

5.1.7.6 Replace Text

defend Uhe eBTU XBMM Pf Uhe bBTUMe

Enter values below for the letters you wish to replace in the text you have entered:

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | a | b | d | e | f | g | h | | | | | | n | | | | | | | | | | | |

Replace

Figure 5.12: Replace text feature

The replace text feature allows users to replace individual letters within a ciphertext that they enter. It works by taking the input text from the analyse text box and then a list of the character entered into the QTableWidgetItem. It works by creating a list of the characters in the uppercase ASCII alphabet and a list of the values within the QTableWidgetItem; empty values are represented as empty strings. The two lists are then zipped together into a python dictionary. The string from the analysis text box is then taken, and if the letter matches a letter key in the dictionary, it is replaced with the corresponding replacement letter value from the dictionary.

This proved to be somewhat problematic as there is a lack of ability to undo the replacements via a button. Currently, if a user wants to change the text back to as it was previously, they must delete the text from the QTableWidgetItem and press replace again which is not necessarily intuitive. This issue was raised during evaluations in the Evaluation chapter.

In order to use the replace text tool to analyse the some of the Zodiac ciphers the researcher attempted to expand on the current method to include a series of ASCII characters which could be represented as a font comprised of the symbols and letters from the Zodiac ciphers created by Largo (2017). However, this proved to become overly complex when using the zipped dictionary method and introduced a lot of errors when replacing text back and forth. This was mainly due to extracting and keeping track of many different characters from the QTableWidgetItem. As such, this particular functionality was removed from the feature as there was not enough time remaining in

the project to either amend this issue or re-write the way in which the functionality replaces text. This was unfortunate as the ability to replace letters within the Zodiac ciphers was an objective the project had hoped to achieve. A better method, if time had allowed, would be to map the symbols to characters within an array or database which could more efficiently handle the characters and keep track of their location without the code becoming overly complex or as confusing.

5.1.8 Testing

5.1.8.1 Ad-hoc Manual Testing

Throughout the development process, ad-hoc manual testing was executed in order to check the tool was functioning as expected. This type of testing is conducted in an informal manner with a view to breaking the system in some way or another. In the case of the Zodiac Cryptogram, the tests focus quite a bit on unexpected input values since the tool is so text dependant.

This testing uncovered several issues which were addressed. For example, in the frequency chart some times of punctuation was not fully removed by the method used, and as a result, the chart was displaying this punctuation. As such, the frequency class was altered in order to limit this.

5.1.8.2 Unit Testing

Unit testing is especially useful when refactoring code as when tests are correctly implemented they provide a method of ensuring the altered functions still work as intended. Additionally, unit tests should in theory catch bugs within the code early on when used at the time of writing the code. Ideally, each unit test should be able to execute independently of other tests.

For the cipher types, several unit tests were created. In the cipher classes, the tests aim to ensure the encryption and decryption method performs as expected - as well as checking that some functions will not accept certain values. For example, in the SimpleSubCipher class tests attempt to set the key length longer, shorter and including characters that the function should not accept and then asserting that the function has

not changed the key value to that of invalid input. Ideally, if time allowed the inclusion of error messages within the test classes would have provided more information to any future developers. Unfortunately, due to time restraints, full test coverage was not achieved within the application.

5.1.8.3 GUI Testing

Ideally, if there had been more time available, then testing of the GUI would have been executed. PyQt5 comes equipped with classes designed to assist in testing GUIs created through Qt. QTest is a class which provides methods to simulate user interaction such as mouse clicks and keystrokes. For example, `QTest.mouseClick(my_button, Qt.LeftButton)` can be performed in a unit test and then the expected outcome can be asserted. This method of testing can be applied to many of the `QWidgets`, which can be implemented to provide thorough test coverage.

5.1.9 Deployment

In order to deploy a PyQt5 application, there are a few options for packaging the final software. The `fman` build system, `PyInstaller`, `cx-freeze` all allow for the deployment of PyQt5 applications to multiple platforms. To create an executable of the Zodiac Cryptogram tool, the researcher used the `fman` build system and `Pyinstaller` separately but ran into an issue with both. An executable was created but unfortunately had a problem with running - more time would be required in order to troubleshoot the issue to produce a working executable file. However, a short readme file was produced detailing how to run the Zodiac Cryptogram tool from the command line.

5.2 Implementation Conclusion

Time constraints were definitely an issue within the development phase, and ideally, a high completion rate of user stories would have been achieved. The architecture of the Zodiac Cryptogram tool was implemented quite closely to the intended design. The visualisations which were implemented perform as expected and offer an interactive

Chapter 5. Implementation

method of cryptanalysis ciphertexts. In terms of the test coverage, this is an underdeveloped part of the project. Again this was down to time. It could be said this was due to subpar implementation of the Hybrid methodology, as not enough time was allocated to the testing process. An agile approach using Test Driven Development may have mitigated this issue by completing the testing at the same time as writing the code.

Chapter 6

Evaluation

6.1 Evaluation Design

Ideally being able to compare test scores between two years of a cryptography class, such as evaluations performed by Luburi et al. (2016) and Adamovic et al. (2018), would have offered a comprehensive method of comparison. However, this is a short experiment with a limited timescale, and the time of year made sourcing students to take part rather tricky. Accordingly, the researcher determined that a comparative evaluation between two groups of students would give suitable results for analysis. Interviews with the students were used to gather qualitative feedback on the experience of using the Zodiac Cryptogram tool. Also, to assess the user interface design, a heuristic evaluation was conducted.

6.1.1 Comparative Experiment

In order to generate quantitative data to assess the performance of the software as an educational tool, an observation-based comparative experiment was created. A similar method of comparison was conducted by Maeref and Algali (2015) to assess their cryptography learning tool 'CPAnim'. The test scores of two groups of students were compared both before and after studying with a cryptography learning tool, one group using CPAnim and the other CrypTool2. These two sets of results were then compared in order to assess the learning increased gained from each tool. To achieve a

similar comparison, a controlled comparative experiment between two tools was thought to be the best method.

In the experiment, students who had previously studied cryptography, are individually asked to complete two tasks using either the Zodiac Cryptogram tool or a website with similar functionality and information available. The website selected to use as a comparison was practicalcryptography.com. The reasons for this are that it is easy to navigate and use, has similar functionality in terms of analysis capabilities and would provide the closest available match in terms of functionality to the Zodiac Cryptogram. In order to give fair comparison participants were asked only to use the n-gram, letter frequency, chi-squared and index of coincidence tools on the website so as to match those which have been implemented in the Zodiac Cryptogram tool.

For all of the student participants, the experiment was conducted within a computer lab (classroom setting) with which they were familiar. For the experiment, only the researcher and the current participant were present in order to minimise distractions and reduce the addition of uncontrollable variables.

Participants were given time to familiarise themselves with their assigned tool. After which they are asked to attempt to decrypt a simple substitution cipher using the analysis tools available to them. Full details of the task can be found in the Appendix . The plaintext chosen was *'When the clock strikes ten, open the door'*, which was then encrypted to *'KHLW OHL DAQDN IOMJNLI OLW, QYLV OHL UQQM'*.

6.1.1.1 Analysis of Comparative Evaluation

The results of the comparative evaluation are summarised in Table 6.1. Participant progress was recorded in two-minute intervals until completion of the ciphertext decryption or 20 minutes had elapsed. For each participant, the plaintext letters they had decrypted by each of these time intervals. The percentage completion was calculated as a ratio of letters decrypted over the total number of ciphertext letters (40). For example, solving 'E' would be counted as not as one letter solved but the number of times 'E' occurred in the text, in this case, 6, giving a completion of 15% for 'E' alone and 2.5% for D.

[illegible]

Table 6.1: Graph of percentage of ciphertext solved by time elapsed, listed by participant identifier and grouped by tool used.

80% of the ZC group found the letters 'THE' within the first four minutes compared to the PC group where only 40% managed this. This would suggest that there could be less of a learning curve, or an easier to use layout to the Zodiac Cryptogram allowing participants to successfully use the tool faster than the Practical Cryptography website.

Of the four that completed in the Zodiac Cryptogram (ZC) group the average time to complete was 15.5 minutes, and for the two that completed in the Practical Cryptography (PC) group the average was 19 minutes - for a difference of 3.5 minutes. Clearly, this is not the most useful measure due to the difference in completion rates within 20 minutes. It should be noted that participant G in the PC group did decrypt the cipher fully before 22 minutes had elapsed, which would have increased the completion rate had it been under a minute sooner.

Within the 20 minute time frame, there was a higher completion rate from the ZC group with 80% completing compared to 40% in the PC group.

Interestingly we can see that for both tools 80% of the participants decrypted the trigram 'THE' and the other 20% decrypted 'E', the most common English letter which was a straight swap for the most common ciphertext letter, followed by 'TH'. This indicates that for both tools, the n-grams were the most likely starting point - followed by the most common letter frequencies.

6.1.2 Observation

As the evaluation took place, the researcher observed the participants and well as recording their screens as they worked. These observations provided several insights. Persson (2004, p. 22) recommends the use of observation in the qualitative evaluation software as a way to capture context, be more discovery-oriented and gain additional insight outside of what participants discuss in interviews. Persson does note that this method runs the risk of participants altering their behaviour in order to meet perceived expectations they believe the research may want. During the task created, it appeared that most participants forgot the observation was taking place for the most part. As such, the risk for this seems to have been mitigated through the task selection. Many participants became very focused on completing the task, a few voiced opinions about

the tools ranging from positive notes about the usefulness of certain charts to frustrations at some design choices. These were all discussed in more depth with participants during the interview process in the following stage.

Summary of Notable Observations:

Two of the participants had issues with using the slider on the *Learn Caesar* page. While only taking them a couple of seconds to correct this seems to be related to the size of the slider being slightly too small. Additionally, it may be due to the slider requiring the user to click either directly on its handle or within the slider bar rather than the area surrounding it. It should be noted that these participants completed the task using a trackpad which may have added to the difficulty of selecting items.

Participant D appeared confused by how the key for the simple substitution example in *Learn* worked. They attempted to click on individual letters in order to 'set' a key via that letter, not in an attempt to change it. It seems that making the implementation of this clearer in some way would be beneficial - perhaps making it possible for the user to edit the key and autofill the remaining letters not used in the key. This participant also stated they were confused by the change between Relative Frequencies and Letter Counts since the charts were too similar.

It should be noted after the task was complete and the full plaintext answer revealed; participant A mentioned they were unaware of the word 'strikes' or the phrase 'clock strikes' as English is not their native language, which could have hindered their ability to solve the cipher. This was lack of foresight on the part of the researcher, and perhaps a more universally known plaintext would have yielded better results.

Participant H had the most difficulty with the task, often becoming confused as to which input box they had been using within the PC website. It seemed that switching pages to compare between the ngrams and common English letter frequencies resulted in them becoming frustrated.

6.1.3 Questionnaire

A short questionnaire was given to those who used the Zodiac Cryptogram tool. Participants were asked to rate the following statements in a range of 1. strongly disagree,

Chapter 6. Evaluation

2. disagree, 3. neither agree nor disagree, 4. agree, 5. strongly agree, in relation to the task they had just completed.

Q1. The learn pages helped me better understand the cipher

Q2. The analyse page helped me better understand cryptanalysis

Q3. The tool helped to identify areas I did not understand within cryptanalysis and cryptography

Q4. The visualisations helped me understand the cryptanalysis

Q5. The Ngrams tool helped me understand the cryptanalysis

Q6. The I.o.C. tool helped me understand the cryptanalysis

Q7. The Frequency tool helped understand the cryptanalysis

Q8. The Replace Text tool helped me understand the cryptanalysis

Q9. The Chi Chart tool helped understand the cryptanalysis

The results from the questionnaire showed a very slight difference between the mean of Q1 - 4.40 and the mean of Q2. 4.60. Both are very positive results; there is a very slight indication that the analyse page proved more useful than the learn pages. This would be expected as the analyse page provided much of the functionality required to solve the cipher.

The I.o.C. and Chi Chart scored equally with a mean of 3.60; again, this would be expected as neither tool would provide a distinct aid in solving the given ciphertext. Q4, Q5 and Q7 all received a mean response of 4.60, indicating the Ngrams and Frequency were equally useful to participants, and the visualisations were considered beneficial to understanding the cryptanalysis. Q8. had the highest standard deviation at 0.89, again this is not a notable difference although it does suggest some found the replace text feature more useful than others.

Overall this questionnaire did not provide a significant amount of insight. The researcher theorises that participants may have answered more positively due to the scaling of 1-5 being somewhat limiting. Also, the statements could have been phrased or selected better to allow for more meaningful comparison. The observation and inter-

views conducted produced results which were more informative in terms of evaluation. The full results and statistical analysis of this questionnaire can be seen in Appendix D.1.

6.1.4 Interview Comments

Interviews were conducted individually with participants after the comparative evaluation had been completed. A standardised open-ended interview approach was used for these interviews. This approach is where a set of predetermined questions are asked during the interview - with the view to having reduced variation in the questions answered by participants. Ensuring the questions asked remain as similar as possible helps to produce answers which are easier to analyse and compare between interviewees. The interviews have been split by group. The interview was conducted with both groups to see if any comments on the design of the Practical Cryptography website might apply to the Zodiac Cryptogram tool. As there were quite a few responses a selection of the more insightful comments have been explored below;

Zodiac Cryptogram:

1. How did you find the experience of using the tool?

Participant C: "I felt engaged in the presented topic, I think that visually, it was clear and easy to use. It helped to visualise the processes of the ciphers in an easy to understand way. There is plenty of information, and it feels like everything is following a logical flow, but jumping from one page to another is problematic and troublesome."

Participant B: "The tool was very intuitive to use - the navigation was clean and flowed well. The visualisations really helped me picture how the different ciphers and cryptanalysis styles worked, and the text helped me understand what each did and how to use each tool."

Several participants noted tool had a good flow and was clear in its layout. Both the participants above mentioned the visualisations being useful to understanding the cryptanalysis and cryptography. The visual elements of the Zodiac Cryptogram tool had hoped to provide a clearer understanding of cryptography and

cryptanalysis to users. The consensus from the interviewees seems to be that this has been achieved, with four out of five participants mentioning their benefit in this regard.

Participant C's comment about the pages being an issue when jumping back and forth was in relation to the tabs used in the analysis page. This sentiment was echoed by other participants who mentioned this made the process of solving the cipher more confusing. Adjustments to the tool should be considered to make the layout more intuitive for this process; this is discussed further in the analysis of the other questions.

2. What did you find most useful in helping you tackle the cryptanalysis section of the study?

All participants replied that they found the frequency and ngram tools most useful, which is to be expected as these are the tools which would aid someone in decrypting a simple substitution cipher. Most stated the ngrams were the most useful of the tools provided.

Participant B : "The tools such as the n-grams and frequency helped me the most, I found that looking for common patterns was really helpful in identifying the most common letters and phrases. Then the replace tool helped me visualise the progress I had made and start to see other patterns."

Participant C: "... the replacement tab was useful for tracking the progress of the task. The n-grams was useful because you could find repeating ngrams which make it more likely you will find the solution to this ngram. The frequency analysis was also useful because you could try out a couple of solutions and see if it fits into the cipher. The ngrams were the most useful because in the cipher there were repeating bigrams that helped me to find a starting point."

As was observed within the comparative results in Table 2.1, the majority of participants first decrypted the ngram 'THE'. Through observation, it was noted that all of the participants consulted the ngrams tool before making this change. This would indicate that even if the participants did guess the ngram from obser-

vation alone, the tool aided them in confirming their theory before committing the change.

3. Is there anything you would change about, or add to the tool?

Participant A: "To give more suggestions – the most frequent letters be suggested at the beginning. See the replace tool on every page. Always be able to see the changes made between each edit."

Participant B: "The only thing that comes to mind to add, and it's just extra, is a way to see the work in the 'replace' tool as you're looking at other tools, so you can see the current state of the cipher you're breaking and not just the original."

In the interviews two participants mentioned they would like more hints as to what to change first. Participant A felt having straight swaps for the most common letters in the ciphertext, and the expected English frequencies suggested would be useful. It would have merit to suggest this along with potential ngrams to the user, as we see in Table 2.1 only one user successfully found a plaintext letter within the first two minutes of starting the task. It should also be considered this was after the initial 5-10 minutes allowed to participants in order to familiarise themselves with the tool. As such, an initial suggest may have pointed them in the right direction - in this case, the most frequent plaintext letter was a straight swap for the most frequent ciphertext. This could be a useful addition, although it would have to be implemented with caution since straight swaps for multiple letters are rarely entirely correct. Thus, it would be important to avoid misleading the user.

Comments about being able to better track changes were common in the interview. This became an apparent issue when at times people forgot which letter they wanted to change when switching between tabs. In essence, there are two main problems; 1. users struggling to remember what changes they made over time, especially if they made an incorrect guess and wanted to change it, and 2. users trying to remember what letters they were considering changing after

analysing the text on another tab. The immediate design change that would help with these issues is moving the replace text tool from a tab to a constantly visible location within the *Analyse* page. A method of tracking the history of changes would be ideal, potentially if the replace text feature would add a new line for each committed change and within that highlight the letters which were switched.

Participant E: "I'd like to be able tell which pages I'm on."

This is an issue which became evident in some of the observation, as there is no change in button colour or other indication of what page a user is on it can be confusing to some if they forget where they are. At present this only appears as a moment of confusion since they are only a few pages within the application but if it were to scale it would be a severe issue to the usability of the tool and should be addressed in future iterations.

4. How do you feel the tool helped you learn about cryptography or cryptanalysis?

Participant C: "I think that because I was engaged I was curious to try the different tools and I wasn't put off by the complexity of the task – I enjoyed using the different tools to break the cipher."

Overall participants mostly responded that the tool helped them understand frequency or ngrams more. This, again is expected since the task required the use of these particular tools the most, meaning participants would have more exposure to these concepts than the others covered in the tool. The comment made by participant C is encouraging to see that the analysis section of the tool was enjoyable to use. When asked if they were engaged by the task or the tool, the participant replied 'both'.

5. Any additional comments or feedback?

Participant B: "... I felt the 'learn' section gave a good overview over the types of ciphers, and the 'analysis' section really helped walk me through different ways to crack the code that I wouldn't have necessarily thought of, and it helped to have it visualised."

The comment from participant B was useful to confirm that the visualisation aspects of the tool were of use to the learning and cipher decrypting. Another participant mentioned they would like the tool as a mobile application so they could use it on the go. Potentially this could be a future avenue for exploration.

Practical Cryptography Website:

1. How did you find the experience of using the tool?

Participants G, H, I and J all mentioned it was hard to keep track of changes made to the ciphertext when using the tool. This is to be expected as the functionality of the ngram and frequency tools was on a separate page to the most common English frequency and ngram page. This somewhat mirrors the issue of struggling to analysis ciphers when tools are not all quickly visible, much like the comments on the switching between tabs in the Zodiac Cryptogram tool. It is likely that since cryptoanalysis can be such a strenuous mental activity remembering additional information such as letter positions and combinations at the same time can be difficult and frustrating. Finding a method to minimise the strain while having a clear and uncluttered UI is an important balance to strike in the design of tools such as these.

2. What did you find most useful to helping you tackle the cryptanalysis section of the study? All participants said they mainly used the n-grams and frequency page, and common English n-grams page.

Participant H: "...It would have been better if when I input my own encrypted phrase if the tool had presented me with a similar graph of the most frequent letters in order so that I could better match up possible answers."

3. Is there anything you would change about, or add to the tool?

Participant G: "More colour – everything is the same, so nothing jumps out at you – more of a contrast between the graphs. The ones that are more frequent could be in a different colour so you can skim it easier. "

Participant J: "I would change the tool by consolidating the actual tool elements onto one page. I would also prefer if the tool ordered the frequency of letters in

my encrypted phrase better to allow for faster letter mapping. Also, if the section on most common words/phrases was ordered alphabetically, that would also help with the mapping process.”

4. How do you feel the tool helped you learn about cryptography or cryptanalysis?

Participant H: ”The background information provided by the tool helped with understanding cryptanalysis as did the frequency of the bigrams. However the section on chi-square analysis was overly complicated and didn’t help me understand cryptanalysis any further”

The Practical Cryptography website does have a rather in-depth page on Chi-squared which contains a lot of equations and can be somewhat intimidating when approaching the subject. It is unlikely it would have helped to any great degree in the given task but knowing that it would not be of much help is also an important piece of information to know. Giving clear information about what a tool can and cannot be of use for upfront and early on would be an excellent principle to integrate into the Zodiac Cryptogram tool.

5. Any additional comments or feedback?

Participant H: ”I have no further information other than I hated everything about the task - Automate it”

While this comment may have been made in good humour, it does point out that many of these tasks can be automated further, removing the more manual side of the cryptography. For the Zodiac Cryptogram, the more manual process was used in order to build a stronger foundational knowledge, much as theorised by Pellone (1995) that repetitive actions help to cement learning. However, having the option to also break these ciphers through a visualised automated means could be an area of future expansion.

It appears that both tools lack a proper method of keeping track of changes. User testing earlier on in the design process would have potentially brought this to light sooner as it is definitely an essential aspect for aiding cryptanalysis. The replace text feature seems to have been at least somewhat useful in helping users keep track, but

there is certainly room for improvement in terms of possibly creating a history of changes and more control over undoing and redoing actions.

It should be considered that the layout of the practical cryptography website was likely not designed for attempting to break anything more than very simple ciphers. The website likely only had a goal of demonstrating how the process works as opposed to functionally utilising the cryptanalysis tools.

Overall it can be said that a lot of useful feedback was gained from the interview process. Mainly issues with the UI interface were identified, such as the annoyance of having to switch back and forth between tabs to access the replace text tool and issues keeping track of progress when trying to decrypt ciphers.

6.1.5 Heuristic Evaluation

Heuristic Evaluation is a commonly used method of assessing the user interface of a piece of software. Nielsen laid out ten key principles to aim for when designing user interfaces, despite being written over twenty years ago they are still in use today. As such, to appraise the UI design and implementation of the Zodiac Cryptogram software, the researcher conducted an evaluation based on each of Nielsen's heuristics.

6.1.5.1 Heuristic Evaluation Summary of Results

1. Visibility of system status

Aim: Inform the user of what is happening inside the system and give feedback on actions within a reasonable time.

Observed: Overall actions all result in an instant or near-instant response. The chi chart visualisation is perhaps the one exception where there is an occasional delay of about 0.5-1 seconds between a change in the text input to the display on the chart updating. This action does result in a noticeable latency at times, which would technically violate this aim. Hence this issue would be fixed in the next iteration if possible.

2. Match between system and the real world

Aim: To have similar language and concepts that would be used in the real world reflected in the system. Information should follow the users real-world experiences and logical order of presentation, taking into account the target audience and their expectations.

Observed: The use of words such as 'encrypt' and 'decrypt' on buttons would be logical to the target audience - students studying cryptography. As they will have encountered this language frequently during their studies, it also succinctly describes the effect that button will have. The terms used such as 'Ngram' and 'Chi-Squared' on the tabs are terms which a student studying cryptography will have come across. There is a logical order of pages, going from 'learn' to 'analyse', which is similar to the learning progression that would be used in a classroom. Therefore, there is a reasonable match between the UI and the real world.

3. User Control and Freedom

Aim: Providing the user with a degree of freedom to control within the system. Often includes the ability to undo or redo actions.

Observed: There is a lack of an option to undo and redo actions in a natural fashion. For example, in the replace text feature, a user could undo the replacement of a letter by altering the assigned text back to as it was previously. There is no button to undo and redo this action quickly in one step, which would be an advantage here.

4. Consistency and Standards

Aim: The user interface should not confuse users through inconsistent use of terminology or graphics. There should be no ambiguity over whether actions or words within the system share a meaning.

Observed: The layout between pages is relatively consistent, with informative text on the left-hand side and interactive elements almost entirely on the right. In terms of words used, they are also consistent; for example, 'encrypt' and 'decrypt' always used in the same way. There is also a consistent use of colour and design style throughout the tool.

5. Error Prevention

Aim: Good design can prevent errors from occurring from user actions. Consequently, the design should aim to remove error-prone conditions or check for them and inform users of this before they commit an action that may produce an error.

Observed: While there are some error prevention methods, this is an area that the software could benefit from further implementation. An example of this would be within the text replace feature. If more than one value is entered an error message in plain English is displayed and only the first value entered is assigned to the letter.

6. Recognition rather than recall

Aim: This principle aims to reduce the mental load placed on users while operating the interface. Users should not have to rely on their memory of instructions or information. Actions and instructions should be visible or easily accessible at all appropriate points.

Observed: There is an issue where the replace text option is located on a separate tab to other analysis tools such as 'frequency'. It would be useful for both tools to be visible and useable at the same time. Otherwise, it relies on the user remembering which letter they wanted to change, which can be bothersome.

7. Flexibility and Efficiency of use

Aim: Having the option to speed up the interactions for an experienced user, while still catering to novice users. This is similar to options such as keyboard shortcuts or advanced controls which are kept out the way for the average user that might find them intimidating.

Observed: In the system, there are not many 'advanced' options. However, options such as showing only duplicate values in the Ngram analysis and the option to switch between 'Relative Letter Frequency' and 'Letter Counts' may be considered an advanced option to some. As such, the system does not hide these

and leaves the option visible to all users. Thus, the interface may be improved further by placing these options elsewhere or in an additional options dialogue or widget, which should provide a more precise and more focused experience.

8. Aesthetic and minimalist design

Aim: Only display information which is relevant to avoid distracting from the key required information or instructions.

Observed: The design does, for the most part, only display relevant information to the page or tab it is displaying. Where the displayed text will change depending on the currently opened analysis tool; however, it can be argued that the design could be minimised further by having the option to hide and show this text as desired.

9. Help users recognize, diagnose, and recover from errors

Aim: When error messages are used, they should appear in plain and easy to understand language, which clearly states the issue and suggests solutions.

Observed: As previously mentioned, there are a few error messages which appear in plain English. There is the potential for additional error checking to be employed, for example in when encrypting and decrypting inputted text in the simple substitution cipher section of Learn once the user has encrypted or decrypted their inputted text both input boxes will be filled with text. To continue using the features, the user can change either text or generate a new key - however, may not be immediately apparent this is possible. Hence an appropriate informative message may be useful here.

10. Help and Documentation

Aim: Ideally, a system could be used without documentation, but in some situations, it is necessary to provide help. In this case, the documentation should be easy to search, list easy to follow steps and not be overly long.

Observed: There is an adequate amount of descriptive text on the function of the analysis tools. However, the addition of a help button with more direct

instructions could improve this.

6.1.5.2 Heuristic Evaluation Conclusion

As stated earlier, the user interface is a crucial component for any learning software. In whole, the Zodiac Cryptogram tool has achieved good coverage of Nielsen's heuristics. However, there is a notable room for improvement. When combined with the feedback from the interviews there are a few clear areas where changes need to be made; keeping track of progress within the application - both in terms of page location and progression of changes to a ciphertext being an obvious one. If the project were to continue these issues could be added to the development backlog to fix or improve them in future iterations.

6.2 Evaluation Conclusion

The evaluation process has brought to light a number of areas for improvement within the tool and several potential new features. Especially within the interview section participants provided several insightful opinions, often out of frustration at either tool, which has been useful in identifying problem areas. In the proceeding chapter, the overall evaluation of both the Zodiac Cryptogram tool, the aims and objectives of the project, and the potential future areas of progression for the tool are discussed and concluded.

Chapter 7

Conclusion & Recommendations for Future Work

7.1 Conclusion Introduction

To conclude this project, the following section will consider to what extent the research questions have been fulfilled and in turn, the overall project aim. After which potential future areas of development will be identified and discussed alongside recommendations for any future researchers who explore this area of educational cryptanalysis software.

7.2 Fulfilment of Research Questions & Project Aim

In order to assess if the project aim has been met, it is first imperative to assess whether our research questions have been adequately answered.

1. Explore how existing software is used to teach cryptography and cryptanalysis at a university level.

The subject of existing software for teaching cryptography at university level was thoroughly explored in the Literature Review section of this project and further explored through client discussions during the requirements gathering stage. An in-depth comparison of existing tools and the results of their evaluations as published was conducted. From this it was discovered that a number of universities

Chapter 7. Conclusion & Recommendations for Future Work

have trailed the use of cryptanalysis learning software within their classes and of those that were considering in the project all indicated a positive effect on the learning attainment of students.

2. Investigate design elements which make these tools effective or ineffective at aiding student learning.

Multiple design elements were identified in the literature review. Several of which were utilised in the design and constructions of the Zodiac Cryptogram tool. Additionally, the evaluation conducted of the tool produced throughout this project can be considered a further investigation into the effectiveness of these elements as the tool incorporated many of them.

3. Develop an e-learning tool to aid university students in learning classical cryptography and cryptanalysis.

The Zodiac Cryptogram tool, while still having room to be expanded and refined, has successfully proven through evaluation that it has aided students in learning classical cryptography and cryptanalysis.

4. Evaluate the created software in terms of ease of use, effectiveness as a learning tool and its overall usability.

The comparative study demonstrated a noticeable increase in ciphertext solving time. Alongside the questionnaire in which all five participants who had previously studied cryptography at university level indicated they felt they had learned more about cryptography and cryptanalysis from the tool.

In summary it can be said that all of the research questions have been answered over the course of this project. Thus, the overarching aim of the project can be considered: Develop an educational cryptanalysis tool which aids the learning process for students and can be used to analyse a selection of ciphers interactively.

Zodiac Cryptogram is an educational cryptanalysis tool and it can interactively analyse a selection of ciphers. It may not have fully achieved all the functionality initially hoped for, especially in term of cipher range, yet it has been proven to be a

useful tool for cryptanalysis and learning all the same. The tool's ability to aid learning was demonstrated in the evaluation of the tool. As such, the overall aim of the project has been achieved.

7.3 Recommendations for Future Work

The tool still has much scope to be expanded upon and improved. Three main areas in which the development could proceed to implement or improve upon have been identified and are explained below;

1. Expanded cipher and cryptanalysis options

There is a lot of potential for expanding the tool as it currently stands by including more cipher types for users to learn about. Many of these have already been mentioned in the literature review and from the earlier questionnaire there is a good prioritised list of user stories to implement in this regard.

Additionally, increasing the cryptanalysis tools would be of benefit to users by increasing the range of concepts for them to learn about via the tool. However, if the tool were to be expanded the current *Analysis* page would have to be reconsidered, partly due to the potential for over-cluttering the GUI. The structure of the tool would also need to be reconsidered if it were to be significantly expanded as currently the main pages are loaded via a `QStackedWidget` which may not be suited to more than ten pages or so.

Most notably, the ability to fully analyse all of the known Zodiac Cryptograms in situ would be the natural progression for this tool. In order to cohesively bring this tool closer to this, it would be essential to add a Learn section about Homophonic substitution ciphers. Also allowing for the easy replacement of multiple ciphertext letters for one or more plaintext letters. In order to execute this in a more elegant way creating functionality of an array or even a database in the back-end of the tool to store and retrieve values would be a better option. At present, the replacement of text is handled by the use of a zipped dictionary within the `replace` text class - which does not expand well to including multiple additional values.

2. Interaction

As it stands, there is some level of interaction within the Zodiac Cryptogram tool. As that may be, there is much further potential for developing the tool to create a more intuitive and free-flowing user interaction experience. Firstly more interactions within the charts, for example, the ability to drag and drop bars in the graphs to rearrange them as best suited to the user. As noted by one participant in the evaluations being able to order the letter frequency values is extremely useful when attempting to compare observed values to expected values. As mentioned in the implementation section, adding the ability to see values on mouse-over of bars within charts would be useful.

One of the main interactions the researcher believes would have been a key addition is that of the ability to highlight sections within the ciphertext. The addition of this highlighting ability and being able to keep notes of potential plaintext solutions and being able to highlight other instances of the highlight section i.e. highlighting an ngram and having the highlight apply to all other occurrences of that particular ngram.

3. Narrative

During initial client discussions, it had been considered that more of a narrative element be included within the project - based on the investigation surrounding the Zodiac Killer. This would be an interesting area to expand the research into and consider the effects of narrative upon learning retention. As noted in the literature review, Hick et al. (2012) had described an event which used a narrative story to help engage students in cryptography. While not included in the tool itself, it was used in conjunction with the CryptTool software. Perhaps the most challenging aspect of this would be developing a balance between narrative and learning without overwhelming the users with information in some way. The addition of a Zodiac narrative element would be the natural progression to round out the tool and give it more depth.

Appendix A

Full Requirements Gathering Questionnaire

Default Report

Questionnaire to Inform a Cryptanalysis Learning Tool
August 18, 2019 9:28 PM MDT

Q1 - Please rate how easy you found understanding the concepts behind classical

ciphers:

| # | Field | Minimum | Maximum | Mean | Std Deviation | Variance | Count |
|---|---|---------|---------|------|------------------|----------|-------|
| 1 | Please rate how easy you found understanding the concepts behind classical ciphers: | 2.00 | 5.00 | 3.86 | 0.99 | 0.98 | 7 |

| # | Field | Choice Count |
|---|----------------------------|-----------------|
| 1 | Extremely easy | 0.00% 0 |
| 2 | Moderately easy | 14.29% 1 |
| 3 | Slightly easy | 14.29% 1 |
| 4 | Neither easy nor difficult | 42.86% 3 |
| 5 | Slightly difficult | 28.57% 2 |
| 6 | Moderately difficult | 0.00% 0 |
| 7 | Extremely difficult | 0.00% 0 |
| | | 7 |

Showing rows 1 - 8 of 8

Q2 - Which ciphers did you struggle most with?

| # | Field | Choice Count |
|---|-------------------------|--------------|
| 1 | Caesar | 0.00% 0 |
| 2 | Simple Substitution | 6.25% 1 |
| 3 | Playfair | 18.75% 3 |
| 4 | Hill | 12.50% 2 |
| 5 | Vigenère | 31.25% 5 |
| 6 | Homophonic Substitution | 31.25% 5 |
| 7 | Other | 0.00% 0 |
| | | 16 |

Showing rows 1 - 8 of 8

Q2_7_TEXT - Other

| |
|-------|
| Other |
|-------|

Q3 - Why do you feel you struggled with these in particular?

Why do you feel you struggled with these in particular?

how they worked wasn't clear at first

Took me a while to grasp the maths behind them and also found learning from a whiteboard hard to follow

Found them hard to follow in the resources I was using

Concept was over my head.

Trying to visualize and understand how the patterns worked, or in some cases identifying what patterns to look for when trying to break an otherwise difficult code.

Q4 - Please select from the list or add any of the resources you used to study

cryptography.

| # | Field | Choice Count |
|---|--|--------------|
| 1 | Lecture Slides | 35.00% 7 |
| 2 | Videos (with visualisations of cryptographic processes) | 25.00% 5 |
| 3 | Videos (without visualisations of cryptographic processes) | 10.00% 2 |
| 4 | practicalcryptography.com | 15.00% 3 |
| 5 | learncryptography.com | 10.00% 2 |
| 6 | simonsingh.net/The_Black_Chamber | 0.00% 0 |
| 7 | CryptTool | 0.00% 0 |
| 8 | Other | 5.00% 1 |

20

Showing rows 1 - 9 of 9

Q4_8_TEXT - Other

| |
|--------|
| Other |
| Google |

Q5 - Which did you find to be the most useful to your understanding of cryptography?

| # | Field | Choice Count |
|---|--|--------------|
| 1 | Lecture Slides | 0.00% 0 |
| 2 | Videos (with visualisations of cryptographic processes) | 60.00% 6 |
| 3 | Videos (without visualisations of cryptographic processes) | 10.00% 1 |
| 4 | practicalcryptography.com | 20.00% 2 |
| 5 | learncryptography.com | 10.00% 1 |
| 6 | simonsingh.net/The_Black_Chamber | 0.00% 0 |
| 7 | CryptTool | 0.00% 0 |
| 8 | Other | 0.00% 0 |
| | | 10 |

Showing rows 1 - 9 of 9

Q5_8_TEXT - Other

Other

Q6 - Which aspect of this resource helped you most?

Which aspect of this resource helped you most?

being able to replay parts of the videos over several times to understand the process

lots of information and images were useful

being able to see the encryption process - especially when metaphors were used like bob and alice, paint colours for diffie-hellman

The combination of visualizations and explanations helped me

Being able to visualize the process while somebody walked me through the logic step by step.

visualisation

Q7 - How much do you agree with the following statement "I found learning cryptography enjoyable"

| # | Field | Minimum | Maximum | Mean | Std Deviation | Variance | Count |
|---|--|---------|---------|------|------------------|----------|-------|
| 1 | How much do you agree with the following statement "I found learning cryptography enjoyable" | 2.00 | 5.00 | 3.00 | 0.93 | 0.86 | 7 |

| # | Field | Choice Count |
|---|----------------------------|-----------------|
| 1 | Strongly agree | 0.00% 0 |
| 2 | Agree | 28.57% 2 |
| 3 | Somewhat agree | 57.14% 4 |
| 4 | Neither agree nor disagree | 0.00% 0 |
| 5 | Somewhat disagree | 14.29% 1 |
| 6 | Disagree | 0.00% 0 |
| 7 | Strongly disagree | 0.00% 0 |
| | | 7 |

Showing rows 1 - 8 of 8

Q8 - Rate how useful you find graphical visualisations or representations when learning?

| # | Field | Minimum | Maximum | Mean | Std Deviation | Variance | Count |
|---|---|---------|---------|------|------------------|----------|-------|
| 1 | Rate how useful you find graphical visualisations or representations when learning? | 1.00 | 2.00 | 1.14 | 0.35 | 0.12 | 7 |

| # | Field | Choice Count |
|---|----------------------------|-----------------|
| 1 | Extremely useful | 85.71% 6 |
| 2 | Moderately useful | 14.29% 1 |
| 3 | Slightly useful | 0.00% 0 |
| 4 | Neither useful nor useless | 0.00% 0 |
| 5 | Slightly useless | 0.00% 0 |
| 6 | Moderately useless | 0.00% 0 |
| 7 | Extremely useless | 0.00% 0 |
| | | 7 |

Showing rows 1 - 8 of 8

Q9 - What features would you like to see in a cryptography and cryptanalysis learning tool?

What features would you like to see in a cryptography and cryptanalysis lea...

videos that explain the ciphers

easy to use, lots of information and images

animations or visual explanations, step by step guides

Step by step explanation with some kind of visualization at each step

Practice ciphers, to work on cracking them - could choose their type and their difficulty level. Videos that illustrate the ciphers and how they work as well.

visualisation

Q10 - Do you feel like an educational tool with visual elements explaining the process of cryptography and cryptanalysis would be useful to your learning experience?

| # | Field | Minimum | Maximum | Mean | Std Deviation | Variance | Count |
|---|---|---------|---------|------|---------------|----------|-------|
| 1 | Do you feel like an educational tool with visual elements explaining the process of cryptography and cryptanalysis would be useful to your learning experience? | 1.00 | 1.00 | 1.00 | 0.00 | 0.00 | 7 |

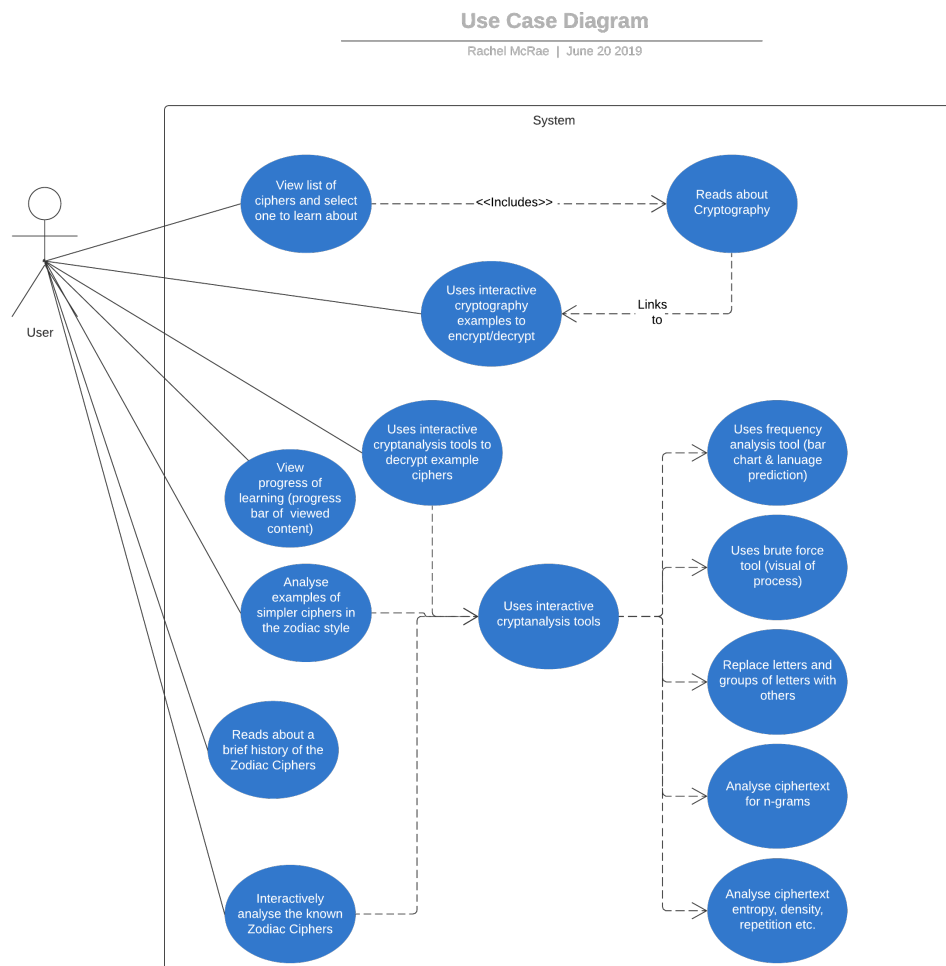
| # | Field | Choice Count |
|---|-------|--------------|
| 1 | Yes | 100.00% 7 |
| 2 | Maybe | 0.00% 0 |
| 3 | No | 0.00% 0 |
| | | 7 |

Showing rows 1 - 4 of 4

End of Report

Appendix B

Use Case Diagram



Appendix C

User Stories

C.1 Implemented

Story: A user can interactively preform frequency analysis on a selection of ciphers.

Details: This may include a bar chart or heat-map (potentially both)

Story: A user can interactively preform n-gram analysis on a selection of ciphers.

Details: This will have a series of n-gram stats such as uniqueness and repeating. These will be visualised and highlighted in the ciphertext.

Appendix C. User Stories

Story: A user can interactively preform chi-squared analysis on a selection of ciphers.

Details: This may include a bar char

Story: A user can interactively preform I.o.C analysis on a selection of ciphers.

Details: This will have suggest a possible language based on I.o.C

Story: A user can read about a selection of ciphers.

Details: This will include their history, examples, and attacks they are susceptible to.

Story: A user can interactively replace letters and/or symbols in a selection of example ciphers in an attempt to decrypt them.

Details: This will likely be done on a line below the ciphertext so as to keep the original ciphertext for reference.

C.2 Selection of Unimplemented

Story: A user can learn about and perform cipher specific analysis for each cipher.

Details: Possible cipher types: Caesar, 'Simple', Affine, Homophonic, Vigenère, Playfair, Hill (chosen for a selection of possible cryptanalysis types)

Story: A user can select different options such as first occurrence, density, entropy by grouping etc. and have them highlight over the ciphertext with stats visualised beside it.

Details: This could include a few different options.

Story: A user can interactively replace multiple letters and/or symbols in a selection of Zodiac ciphers in an attempt to decrypt them.

Details: This will allow for replacing single letters with a variety of symbols and letters in the text, applicable to homophonic ciphers.

Appendix D

Evaluation



Participant Information Sheet for Comparative Evaluation of Cryptanalysis Learn Tool

Name of department: Computer Information Sciences

Title of the study: An exploration of Cryptanalysis through the Zodiac Cryptogram **Introduction**

My name is Rachel McRae, I am a postgraduate student at Strathclyde University conducting a comparative study for my research into creating a Cryptanalysis Learning Tool aimed at university students.

What is the purpose of this research?

The purpose of this research is exploring cryptanalysis and methods of visualising the process to aid learning. Through the research software which is aimed at university students with a view to making the process of learning about classical cryptography easier will be evaluated. The software will focus on visualising a selection of classical ciphers and cryptanalysis methods. It is hoped that through this evaluation we can assess how well it functions as a learning tool.

Do you have to take part?

You are under no obligation to complete this any part of this evaluation or submit your answers, participation is entirely voluntary. Answer whichever questions you feel comfortable with and feel free to leave the rest blank.

What will you do in the project?

You will be asked to complete a short series of exercises using either the learning tool or a website with similar functionality against which it will be compared. The process is detailed on page 4-5 of this document.

Why have you been invited to take part?

You have been invited to take part in this research as you have previously studied cryptography as part of a university module, meaning you are the target audience for this software.

What information is being collected in the project?

From this comparative study the screen recordings of your time using the tool or website will be collected (if you consent to this), observational notes taken by the researcher as you interact with the tool, your responses to questionnaire and the open interview questions will all be collected. Interview questions will also be recorded if you give your consent but you can still participate without being recorded if you wish. If at anytime you want to have any of this data deleted please make the researcher, Rachel McRae, aware either in person or via email (rachel.mcrae.2018@uni.strath.ac.uk) and this will be done as soon as possible.

Who will have access to the information?

Please be assured that your responses will be kept completely confidential and responses will be securely stored in compliance with GDPR guidelines. Only the Principal Investigator, Rachel McRae, on this study will have access to the results of the data collected and any data used in the final report will be anonymised.

The place of useful learning

The University of Strathclyde is a charitable body, registered in Scotland, number SC015263



Where will the information be stored and how long will it be kept for?

Responses and any collected data will be kept for the duration of this project (no longer than 31st October 2019) in a secure data repository and then disposed of. Your name will be securely stored and any data pseudonymised and stored separately from your responses and other collected data. If at any time you wish to withdraw your data from the study you may do so by informing the researcher, all stored data will be erased on or before Oct 31st 2019.

What happens next?

If you are interested in taking part then please read the entirety of this participant information and then indicate your consent at the bottom of the page. You can then indicate to the researcher if you are ready to take part in the study either in person or via e-mail and time for you to take part will be arranged. The researcher will talk you through the process of the study as well as discussing the instructions with you before the study begins. If you do not wish to participate, thank you for your time.

Researcher contact details:

Researcher Contact: Rachel McRae - rachel.mcræ.2018@uni.strath.ac.uk

Supervisor: Dr Rosanne English - rosanne.english@strath.ac.uk

This research was granted ethical approval by the Departmental Ethics Committee.

If you have any questions/concerns, during or after the research, or wish to contact an independent person to whom any questions may be directed or further information may be sought from, please contact:

Secretary to the Departmental Ethics Committee
Department of Computer and Information Sciences,
Livingstone Tower
Richmond Street
Glasgow G1
1XH
email:ethics@cis.strath.ac.uk

- I confirm that I have read and understood the Participant Information Sheet for the above project and the researcher has answered any queries to my satisfaction.
- I confirm that I have read and understood the Privacy Notice for Participants in Research Projects and understand how my personal information will be used and what will happen to it (i.e. how it will be stored and for how long).
- I understand that my participation is voluntary and that I am free to withdraw from the project at any time, up to the point of completion, without having to give a reason and without any consequences.
- I understand that I can request the withdrawal from the study of some personal information and that whenever possible researchers will comply with my request.
- I understand that anonymised data (i.e. data that do not identify me personally) cannot be withdrawn once they have been included in the study.
- I understand that any information recorded in the research will remain confidential and no information that identifies me will be made publicly available.

The place of useful learning

The University of Strathclyde is a charitable body, registered in Scotland, number SC015263



- I consent to being a participant in the project.

Do you consent to audio recording for the interview portion of this study? No ☐ Yes ☐

Do you consent to the use of screen recording software tracking your mouse movements? No ☐ Yes ☐

(PRINT NAME)

Signature of Participant:

Date:

Outline of the Study:

As a participant you will be assigned one of two tools to complete the following tasks. You will either be using the Zodiac Cryptogram Tool or <http://practicalcryptography.com>, the researcher will tell you which before you begin. During this study your screen will be recorded using screen capture software to track your mouse movements and progress, no audio will be captured. The researcher will also be taking notes while observing you complete the tasks, if you are uncomfortable with any of this please let the researcher, Rachel McRae, know. Please feel free to ask questions or leave the study at anytime.

1. Take 5-10 minutes to familiarise yourself with the tool you are using. In this time see if you can read about simple substitution ciphers. If you are using Practical Cryptography you will find them here; <http://practicalcryptography.com/ciphers/classical-era/simple-substitution/>. If you feel happy to proceed to the next task before the 10 minutes are up feel free to indicate this to the researcher.
2. The researcher will provide you with a ciphertext, using only the cryptanalysis tools in either tool attempt to break it. For Zodiac Cryptogram this is on the Analysis page and for Practical Cryptography you can find this here; <http://practicalcryptography.com/cryptanalysis/>. For this task you have 20 minutes but do not worry if you do not complete it within this timeframe.
3. You will now be asked to rate each of the following statements in a range of [1 – strongly disagree, 2 – disagree, 3 neither agree nor disagree, 4 – agree, 5 – strongly agree].

| | |
|-----------|---|
| <u>Q1</u> | <u>The learn pages helped me better understand the cipher</u> |
| <u>Q2</u> | <u>The analysis page helped me better understand cryptanalysis</u> |
| <u>Q3</u> | <u>The tool helped to identify areas I did not understand within cryptanalysis and cryptography</u> |

The place of useful learning

The University of Strathclyde is a charitable body, registered in Scotland, number SC015263



| | |
|-----------|---|
| <u>Q4</u> | <u>The visualisations helped me understand the cryptanalysis</u> |
| <u>Q5</u> | <u>The Ngrams tool helped me understand the cryptanalysis</u> |
| <u>Q6</u> | <u>The IOC tool helped me understand the cryptanalysis</u> |
| <u>Q7</u> | <u>The Frequency tool helped understand the cryptanalysis</u> |
| <u>Q8</u> | <u>The Replace Text tool helped me understand the cryptanalysis</u> |
| <u>Q9</u> | <u>The Chi Chart tool helped me understand the cryptanalysis</u> |

4. If you wish to proceed, you can answer a short interview consisting of the following questions. If you have consented to recording this portion of the interview will be audio recorded.

- How did you find the experience of using the tool?
- What did you find most useful to helping you tackle the cryptanalysis section of the study?
- Is there anything you would change about, or add to the tool?
- How do you feel the tool helped you learn about cryptography or cryptanalysis?
- Any additional comments or feedback?

The study is now finished – thank you for your time

The place of useful learning

The University of Strathclyde is a charitable body, registered in Scotland, number SC015263

Appendix D. Evaluation

| # | Field | Minimum | Maximum | Mean | Std Deviation | Variance | Count |
|---|--|---------|---------|------|---------------|----------|-------|
| 1 | The learn pages helped me better understand the cipher | 4.00 | 5.00 | 4.40 | 0.49 | 0.24 | 5 |
| 2 | The analysis page helped me better understand cryptanalysis | 4.00 | 5.00 | 4.60 | 0.49 | 0.24 | 5 |
| 3 | The tool helped to identify areas I did not understand within cryptanalysis and cryptography | 4.00 | 5.00 | 4.40 | 0.49 | 0.24 | 5 |
| 4 | The visualisations helped me understand the cryptanalysis | 4.00 | 5.00 | 4.60 | 0.49 | 0.24 | 5 |
| 7 | The Frequency tool helped understand the cryptanalysis | 4.00 | 5.00 | 4.60 | 0.49 | 0.24 | 5 |
| 5 | The Ngrams tool helped me understand the cryptanalysis | 3.00 | 5.00 | 4.60 | 0.80 | 0.64 | 5 |
| 6 | The IOC tool helped me understand the cryptanalysis | 3.00 | 5.00 | 3.60 | 0.80 | 0.64 | 5 |
| 9 | The Chi Chart tool helped understand the cryptanalysis | 3.00 | 5.00 | 3.60 | 0.80 | 0.64 | 5 |
| 8 | The Replace Text tool helped me understand the cryptanalysis | 3.00 | 5.00 | 4.00 | 0.89 | 0.80 | 5 |

Figure D.1: Full results from Evaluation Questionnaire

Bibliography

Signals and Slots: Qt Core 5.13.0. URL doc.qt.io/qt-5/signalsandslots.html.

Practicalcryptography.com, 2019. URL <http://practicalcryptography.com/cryptanalysis/letter-frequencies-various-languages/english-letter-frequencies/>.

Scientific Graphics and GUI Library for Python. 2019. URL www.pyqtgraph.org/.

2019. URL <http://zodiackiller.com/FAQ.html>.

Abdrah Abuzaid, Huiming Yu, Xiaohong Yuan, and Bill Chu. The design and implementation of a cryptographic education tool. pages 193–198, 2011. doi: 10.5220/0003301301930198.

S. Adamovic, M. Sarac, D. Stamenkovic, and D. Radovanovic. The importance of the using software tools for learning modern cryptography. *International Journal of Engineering Education*, 34(1):256–262, 2018.

Wasim A Alhamdani. Teaching Cryptography Using Design Thinking Approach. *Journal of Applied Security Research*, 11(1):78–89, 2016. ISSN 19361629. doi: 10.1080/19361610.2015.1069646. URL <http://dx.doi.org/10.1080/19361610.2015.1069646>.

Alex Barter. *Chi-Squared Statistic*. 2017a. URL <http://alexbarter.com/statistics/chi-squared-statistic/>.

Alex Barter. *Index of Coincidence*. 2017b. URL <http://alexbarter.com/statistics/index-of-coincidence/>.

Bibliography

- Craig P. Bauer. *The Zodiac Ciphers: What We Know*. 2017. URL <https://www.history.com/news/the-zodiac-ciphers-what-we-know>.
- Friedrich L Bauer. *Decrypted secrets*. Springer, 1997.
- BinFire. *Hybrid project management flow schematic*. 2018. URL https://www.binfire.com/templates/landing/img/ebooks/hybrid_soft_manage/diagr01.png.
- E.J. Braude and M.E. Bernstein. *Software Engineering: Modern Approaches, Second Edition*. John Wiley, 2016. ISBN 9781478633037. URL <https://books.google.co.uk/books?id=kILL1CwAAQBAJ>.
- CAST. New report confirms agile/waterfall mix produces the best code quality. *CAST Software*, Sep 2014. URL <https://www.castsoftware.com/discover-cast/press-releases/docs/press-releases/new-report-confirms-agile-waterfall-mix-produces-the-best-code-quality>.
- Wang Changji, Liao Dingfeng, and Huang Huajie. The Design and Analysis of Information Security Teaching and Learning Platform. *2009 First International Workshop on Education Technology and Computer Science*, 3:598–602, 2009. doi: 10.1109/ETCS.2009.667.
- M. Cohn. *User Stories Applied: For Agile Software Development*. Addison-Wesley signature series. Addison-Wesley, 2004. ISBN 9780321205681. URL <https://books.google.co.uk/books?id=SvIwuX4SVigC>.
- crypto.interactive-maths.com. Vigenère cipher, 2019. URL <https://crypto.interactive-maths.com/vigenegravere-cipher.html>.
- Cryptool.org. *Cryptool.org*, 2019. URL <https://www.cryptool.org/en/cryptool2>.
- Justin de los Angeles. Project framework comparisons: Agile vs. waterfall vs. hybrid vs. lean, Mar 2018. URL <https://medium.com/@jdelosangeles/project-framework-comparisons-agile-vs-waterfall-vs-hybrid-vs-lean-dc6801d217e4>.

Bibliography

- John Deacon. Model-view-controller (mvc) architecture. john deacon computer system development, consulting & training 2005:1-6. 2009.
- Eric Fouh, Monika Akbar, and Clifford A. Shaffer. The Role of Visualization in Computer Science Education. *Computers in the Schools*, 29(1-2):95–117, 2012. ISSN 07380569. doi: 10.1080/07380569.2012.651422.
- S Hick, Bernhard Esslinger, and A Wacker. Reducing the complexity of understanding cryptology using CrypTool. *IMSCI 2012 - 6th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings*, pages 54–59, 2012. URL <http://www.scopus.com/inward/record.url?eid=2-s2.0-84896447014&partnerID=40&md5=507636665eafd5ba5c9b190653fb571b>.
- Tom Juzek. The zodiac killer’s unsolved z340: not your vanilla substitution cipher, 2018. URL <http://tsjuzek.com/blog/z340.html>.
- Frank H. Katz. A comparison of different methods of instruction in cryptography. In *Proceedings of the 2014 Information Security Curriculum Development Conference*, InfoSec ’14, pages 14:1–14:1, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-3049-7. doi: 10.1145/2670739.2670755. URL <http://doi.acm.org/10.1145/2670739.2670755>.
- Largo. *New Zodiac ciphers font – Zodiac Killer Ciphers*. 2017. URL <http://www.zodiackillerciphers.com/?p=726>.
- Can Li, Jun Ma, Jun Tao, Jean Mayo, Melissa Keranen, and Chaoli Wang. VIGvisual : A Visualization Tool for the Vigenère Cipher. pages 129–134, 2015.
- Nikola Luburi, Milan Stojkov, Goran Savi, Goran Sladi, and Branko Milosavljevi. Crypto-Tutor : An educational tool for learning modern cryptography. pages 205–210, 2016. doi: 10.1109/SISY.2016.7601498.
- Mabroka (Sebha University) Maeref and Fatma (Sebha University) Algali. An Empirical Evaluation of CrypTool in Teaching Computer Science. pages 93–100, 2015.

Bibliography

- M.S. Mayzner and M.E. Tresselt. *Tables of Single-letter and Digram Frequency Counts for Various Word-length and Letter-position Combinations*. Psychonomic monograph supplements. Psychonomic Press, 1965. URL books.google.co.uk/books?id=FI7BHgAACAAJ.
- Rachel McRae. *(unpublished) Project Report: An exploration of cryptanalysis using the Zodiac Cryptograms*.
- D J K Mewhort. Case-sensitive letter and bigram frequency counts from large-scale English corpora. 36(3):388–396, 2004.
- Jakob Nielsen. 10 usability heuristics for user interface design.
- Peter Norvig. *English Letter Frequency Counts: Mayzner Revisited or ETAOIN SRHLDCU*. 2013. URL norvig.com/mayzner.html.
- Gennaro Pellone. Educational software design: A literature review. *Australasian Journal of Educational Technology*, 11(1), 1995. ISSN 1449-3098. doi: 10.14742/ajet.2070.
- Sara Persson. *Qualitative Methods in Software Engineering*. 2004.
- V. Rahimian and R. Ramsin. Designing an agile methodology for mobile software development: A hybrid method engineering approach. In *2008 Second International Conference on Research Challenges in Information Science*, pages 337–342, June 2008. doi: 10.1109/RCIS.2008.4632123.
- Guido Rößling, Vicki Almstrum, U Texas Austin, Wanda Dann, Rudolf Fleischer, Hong Kong U Sc, Chris Hundhausen, Ari Korhonen, Lauri Malmi, Myles McNally, Alma College, Susan Rodger, and U Rey Juan Carlos. Exploring the role of visualization and engagement in computer science education. *SIGCSE Bull.*, 35(2), June 2002. doi: 10.1145/782941.782998. URL <http://doi.acm.org/10.1145/782941.782998>.
- N Paul Schembari. “Hands-On Crypto”: Experiential Learning in Cryptography. *Proceedings of the 11th Colloquium for Information Systems Security Education*, pages 7–13, 2007.

Bibliography

- Dino Schweitzer and Leemon Baird. The Design and Use of Interactive Visualization Applets for Teaching Ciphers. pages 69–75, 2006. doi: 10.1109/IAW.2006.1652079.
- S. Singh. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. Fourth Estate, 1999. ISBN 9781857028799. URL <https://books.google.co.uk/books?id=6ongAAAAMAAJ>.
- Simon Singh. *The Black Chamber - Homophonic Cipher*. 2019. URL https://www.simonsingh.net/The_Black_Chamber/homophonic_cipher.html.
- Richard Spillman. A software tool for teaching classical cryptology. pages 1–9, 2002.
- Feng Yang, Cheng Zhong, Mengxiao Yin, and Yiran Huang. Teaching Cryptology Course Based on Theory-Algorithm- Practice- Application Mode. *2009 First International Workshop on Education Technology and Computer Science*, 2:468–470, 2009. doi: 10.1109/ETCS.2009.366.
- Rong Yang, Layne Wallace, and Ian Burchett. Teaching Cryptology At All Levels Using CrypTool. pages 22–28, 2011.

Bibliography