Enigma Machine Application

“A Cloud-Based Working Simulation for Modern Cryptography”

Oscar Bernal, Jim Esposito, Melony Henderson, Richard Robinson

Computer Information Systems Graduate

Program Night Students

ECPI University

Table of Contents

[Abstract 3](#_Toc315545551)

[Background 4](#_Toc315545552)

[Proposed Work](#_Toc315545601) 6

[Appendix 1: German Enigma Machine Simulator 7](#_Toc315545554)

[1 Introduction](#_Toc315545556) 9

[2 System/Description](#_Toc315545560) 9

[3 Flow Charts & Classes](#_Toc315545573) 12

[4 Conclusion](#_Toc315545573) 15

[APPENDIX A –STORY BOARDS](#_Toc315545596) 16

[APPENDIX B – TEST CARDS](#_Toc315545597) 18

[APPENDIX C – HARDWARE WIRING DIAGRAMS](#_Toc315545600) 20

[APPENDIX D –ROTORS & REFLECTORS](#_Toc315545601) 23

[APPENDIX E –ROTORS & REFLECTORS FLOW](#_Toc315545601) 25

[APPENDIX F – TEAM TWO LOG](#_Toc315545604) 28

[APPENDIX G – CODE BOOK](#_Toc315545604) 55

[Definitions](#_Toc315545611) 60

[References](#_Toc315545710) 61

Abstract

There are several approaches to software engineering. This paper will provide some background in regards to software engineering and the various approaches. It will show a informative account of how Extreme Agile Software Development with pairing as the discipline is used to construct a virtual simulator application of the infamous German Enigma machine. It will also provide an understanding of inner workings of the notorious, almost unbreakable cryptographic German Enigma machine that plagued the allies during World War II.(Sommerville, 2001)

*Keywords:* Enigma Machine, software engineering, extreme agile software development, pair programming, encoding, decoding, encryption, decryption, cipher, cryptography

# Background

Software engineering is an engineering discipline concerned with all aspects of software production. It has been around since 1969 when it was first named and introduced at a NATO conference to address software development problems. These problems consisted of large software systems that were late, did not deliver the functionality needed by their users, cost more than expected, and were unreliable. Because software has permitted us to investigate space and produce the World Wide Web, our utmost noteworthy information system to mankind. We are now confronted with a set of new challenges-climate change and extreme weather, diminishing natural resources, increasing world population that needs to be nourished and lodged, international terrorism and the quality of life for our elderly people. New technologies are necessary to help us tackle these difficulties and, certainly software will play a definite part in these technologies. This makes software engineering a critically essential technology for the future of mankind. All though there are still problems with software projects, over budget and not being completed on time, we must continue to educate software engineers. These complications should not obscure the real achievements in software engineering and the remarkable methods, disciplines, and technologies that have been established. (Sommerville, 2001)

The original Enigma cipher machine was an ingenious piece of technology that was united with military history and the covert world of espionage, code breakers and intelligence. During the Second World War, not ever before has the destiny of so many lives been influenced by one cryptographic machine. The Enigma is the most legendary, alluring, interesting paradigm of the battle between code makers and code breakers; it showed the significance of cryptography to military and civil intelligence. (Psaul Reuvers & Marc Simons, 2009-2012)

Proposed Work

Team Two, consisting of Oscar Bernal, Jim Esposito, Melony Henderson, and Richard Robinson, has been assigned the task of constructing a cloud-based simulator that provides future students with a virtual German Enigma machine that they can send and receive messages to one another. The system will visualize the encryption/decryption process so that students can understand the inner workings of the machine. The software engineering method and discipline to be used is Extreme Agile Software Development with pair programming. The primary documentation required for this method includes story cards, test cards, and progress reports to our client. Developers code/program in pairs, checking each other’s work and providing the support to always do a good job. The simulation will allow for secure login and authentication as well as registration for new users.

Appendix 1: German Enigma Machine Simulator

Introduction

The original German Enigma machine was an electro-mechanical rotor machine. The cipher machine was used to encrypt and decrypt secret messages. Because of the multiple default rotors, reflectors, and plug settings, theoretically, approximately 3,283,883,513,796,974,198,700,882,069,882,752, 878, 379,955,261,095,623,685, 444,055,315,226,006,433,615,627,409, 666,933,182,371,154,802,769,920,000,000,000 combinations were possible. (Miller, 1996-2001)

In 1923, the first cipher machine, Enigma A, was marketed. A big and heavy machine with an incorporated typewriter weighed about 110 pounds. The typewriter was soon replaced by a lamp panel. The Enigma D was introduced and commercialized in several versions with different rotor wirings and sold across Europe to armed forces and diplomatic venues in 1927. The Enigma D version had three normal rotors and one reflector. (Rijmenants, 2004-2012)

System/Description

Team Two (Oscar Bernal, Jim Esposito, Melony Henderson, Richard Robinson)

Application: Enigma Machine Application based on the five default rotors, used by the Wehrmacht and Kriegsmarine and the two wide reflectors Wehrmacht and Luftwaffe. The application utilizes three rotors out of five, one reflector of two, and up to ten plugs. The rotors and reflectors each have twenty-six letters and the plugs simulate a straight patch that changes a letter. (example: A:G means when the A is typed/entered, it becomes a G to the simulator) The programming language used is Xcode. Facebook is being used for logon, registration, and authentication.

Technical notes about Enigma engine:

* Engine was written as a restful API in C# and deployed on a Windows 2008 Server. The database backend was developed using SQL Server 2008. The site is hosted on [Winhost.com](http://Winhost.com/) at <http://www.koinoniasgr.com/enigma/>.  The endpoint to encrypt data is /Encrypt/.
* The API accepts JSON input to encode and sends a JSON response in return.
* XCode was used to write the iPhone user app which communicates with the API to encrypt and decrypt messages.
* Github was used as the repository for both C# and XCode projects as well as other methodology documentation.

Testing:

* To test the machine, several third party sites (<http://enigmaco.de/enigma/enigma.html>, <http://www.terrylong.org/> and [http://www.bletchleypark.org.uk/content/enigmasim.rhtm#](http://www.bletchleypark.org.uk/content/enigmasim.rhtm)) were used to decrypt messages encrypted by our new engine and vice versa.
* Manual testing was done before production online system testing was completed

A code book has been developed to assist the instructor with simulator set up. It is labeled in German and has a minimum of 62 random set-up combinations with additional space to invent alternatives. Once logged into the simulator, the instructor will determine the rotor type, rotor positions, reflector, and plug settings. To encrypt, both parties will need to be set up exactly the same. Send you message and it will be received encrypted. Enter the encrypted message and it will decrypt.

Millstones:

Millstones encountered included our initial testing. When we manually tested the encryption using the default rotors and reflectors settings, we found that we could not get the same results more than once. This caused the team to dig deeper into our research; we were able to get a better understanding of the inner workings of the machine's schematics, how each rotor affected the others, what the reflector does, and how the plugs serve as a patch or switch.

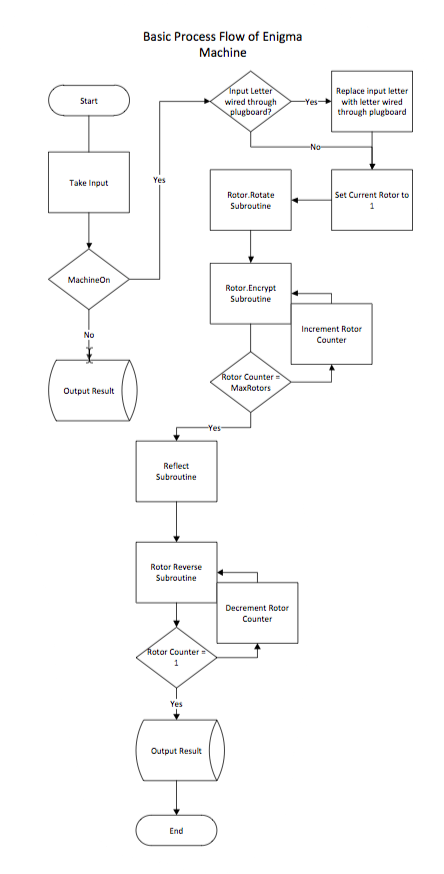
Another millstone was scheduling to ensure we followed the pair programming methodology. The work around for half of the team was to use TeamViewer for remote pairing. We discovered that TeamViewer runs slow (appears to have a delay) on a hotspot, but it will work. The other half did pair programming face-to-face.

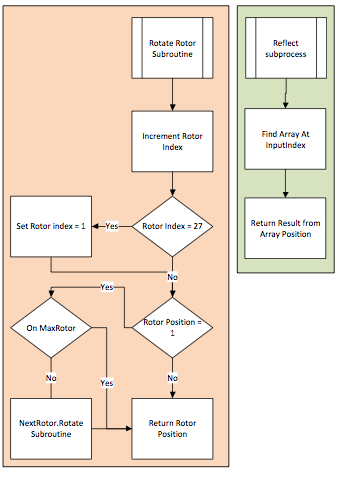
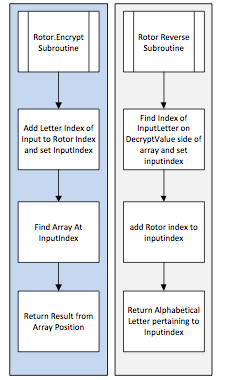
The junior programmers encountered difficulties codding due to inexperience but, the seniors were there to assist and explain as needed.

The plug board proposed a challenge when attempting to get it to function properly in the application. Again, our senior programmers stepped up to the plate to provide guidance , assistance, and alternatives to implement application code.

While no coding was done alone, testing was continuous and done between every change or alteration made during pair programming.

We utilized two senior programmers and two analysts; clocking one hundred fifty-five hours total. For two analysts and two programmers, the cost per hour is $220. Our total project cost to date is $34,100.







In conclusion, we present our observations the about Agile Pair Programming methodology. This methodology may be best suited to an organization employing a larger number of employees that are available in the small team environment. Without the benefit of a larger company, the small team environment would not allow the conflicting schedule factor, thereby causing delays in development. With our small team environment, scheduling conflicts were encountered. To prevent lost productivity, it allowed for more detailed planning and research by the group. Fully functional pseudo-code, class diagrams, and flowcharts were developed as a result of the scheduling conflicts. Additionally, we were able to apply the research and attain an advanced comprehension of the inner workings and design of the enigma machine. In larger, normal work environment, these items are less likely to have been produced to such a degree or in-depth detail due to pair programmers will be available and ready to code. We found that in our inability to pair program, we were better prepared to produce a higher quality product. The created flowcharts, pseudo-code, and class diagrams in conjunction with a more detailed understanding of the rotors, reflectors, and enigma wiring/operation diagrams provided our actual time to pair program extremely productive. With our methodology, we were able to maintain productivity with other teams as well as out produce others. We believe that this methodology would be best suited for your junior programmers and/or your programmers inexperienced in specific languages allowing learning and teaching each other. As a training tool, pair programming a senior programmer with a junior programmer this can be useful sharing knowledge and expertise. As simulated in our small team environment, usually this results with the senior programmer doing most of the actual coding due to experience, time constraints, as well as the junior programmer not being as familiar with code and syntax.

APPENDIX A – STORY CARDS

**Team 2 Story Boards**

**Completed by:** [Oscar Bernal, Jim Esposito, Melony Henderson, Richard Robinson ]

|  |  |  |  |
| --- | --- | --- | --- |
| **Done?** | **Initial Story Card** | **Due By** | **Notes** |
| **Yes** | Planning | 6/19/12 | initial planning meetng |
| **Yes** | Preparation | 6/19/12 | contact information exchange within the team |
| **Yes** | Get specifications from client | 6/21/12 | get with client for project specifications |
| **Yes** | Planning/Preparation | 6/21/12 | team meeting to plan implementation of client specs |
| **Yes** | Planning/Preparation | 6/21/12 | get link for documentation form/update team/submit paperwork |
| **Yes** | 1st Progress Report | 6/28/12 | present oral progress report |
| **Yes** | 2nd Progress Report | 7/3/12 | present oral progress report |
| **Yes** | 3rd Progress Report | 7/12/12 | present oral progress report |
| **Yes** | Present Product | 7/19/12 | present product/presentation/documentation/paper |
|  |  |  |  |

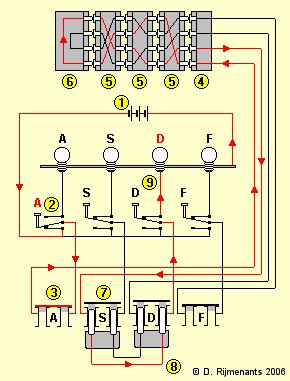
|  |  |  |  |
| --- | --- | --- | --- |
| **Done?** | **Customer Story Card** | **Due By** | **Notes** |
| **Yes** | Planning | 6/22/12 | meet with client to get project specifications |
| **Yes** | client | 6/22/12 | specify deadline/expectations from client/prioritize tasks from client |
| **Yes** | 1st Progress Report | 6/28/12 | follow-up with client/status reports |
| **Yes** | 2nd Progress Report | 7/3/12 | follow-up with client/status reports |
| **Yes** | 3rd Progress Report | 7/12/12 | follow-up with client/status reports |
| **Yes** | Present Product | 7/19/12 | present product/presentation/documentation/paper |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

(example)

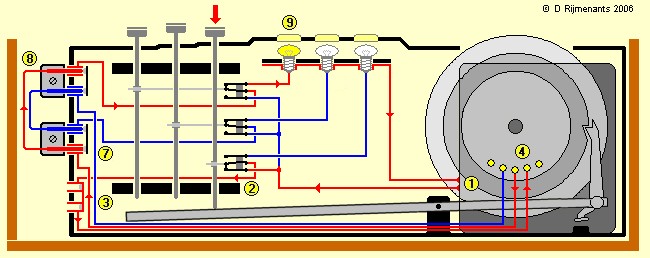
APPENDIX B – TEST CARDS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| **Steps** | **Test Cases** | **Due By** | **Expected Response** | **Pass/Fail** | **Notes/Comments** |
| **1** | click login | 7/15/12 | request to enter user id or create an account | **Pass** | valid user id, login else register |
| **2** | authentication-login | 7/15/12 | enter valid user id and password | **Pass** | valid user password (>6 characters to include 1 letter, 1 number) |
| **3** | authentication-registration | 7/15/12 | create an account | **Pass** | not a valid user; follow instructions to create an account and valid password (>6 characters to include 1 letter, 1 number) |
| **4** | forgot password | 7/15/12 | select forgot password | **Pass** | reset password(>6 characters, 1 letter, 1 number) to be sent to e-mail address of record |
| **5** | valid user id | 7/15/12 | enter valid user id and password | **Pass** | continue to site |
| **6** | invalid user id | 7/15/12 | enter invalid us id | **Pass** | send to registration screen; follow instruction to create an account and valid password (>6 characters, 1 letter, 1 number) |
| **7** |  |  |  |  |  |
| **8** |  |  |  |  |  |
| **9** |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| **12** |  |  |  |  |  |
|  | (example) |  |  |  |  |

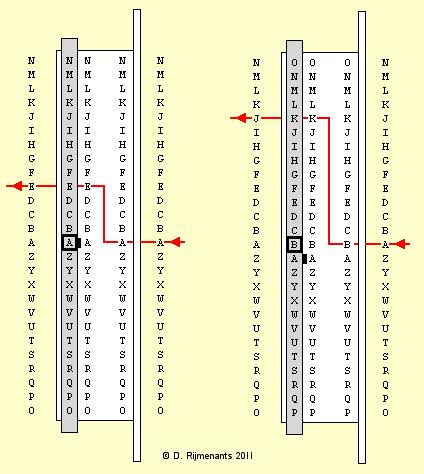
APPENDIX C – HARDWARE WIRING DIAGRAMS



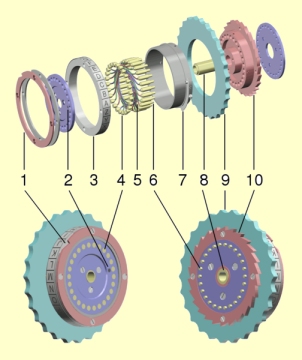
(Rijmenants, 2004-2012)



(Rijmenants, 2004-2012)



APPENDIX D – ROTORS & REFLECTORS



(Rijmenants, 2004-2012)



APPENDIX E – ROTORS & REFLECTORS PATH

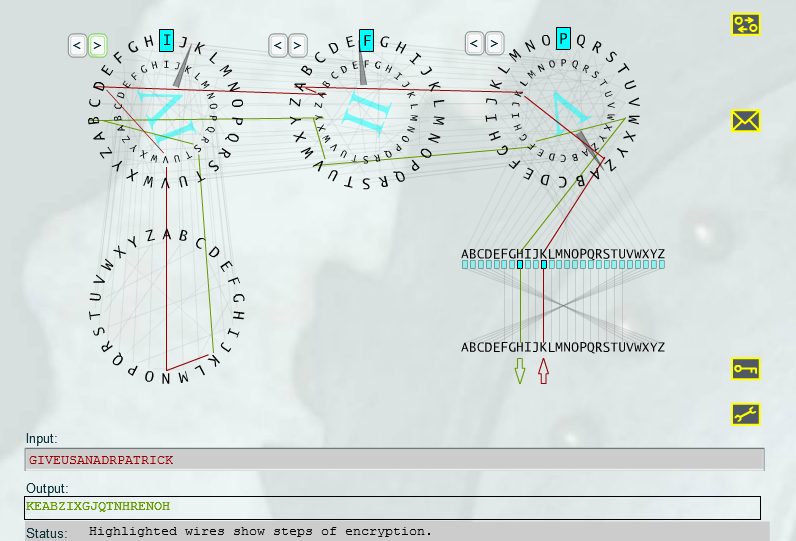
Default rotors, used by the Wehrmacht and Kriegsmarine

|  |
| --- |
| Entry = ABCDEFGHIJKLMNOPQRSTUVWXYZ (rotor right side)  ||||||||||||||||||||||||||  I = EKMFLGDQVZNTOWYHXUSPAIBRCJ  II = AJDKSIRUXBLHWTMCQGZNPYFVOE  III = BDFHJLCPRTXVZNYEIWGAKMUSQO  IV = ESOVPZJAYQUIRHXLNFTGKDCMWB  V = VZBRGITYUPSDNHLXAWMJQOFECK |

Default wide reflectors Wehrmacht and Luftwaffe:

|  |  |
| --- | --- |
| |  | | --- | | Contacts = ABCDEFGHIJKLMNOPQRSTUVWXYZ  ||||||||||||||||||||||||||  Reflector B = YRUHQSLDPXNGOKMIEBFZCWVJAT  Reflector C = FVPJIAOYEDRZXWGCTKUQSBNMHL | |

(Rijmenants, 2004-2012)



(Spiess, 2009)

APPENDIX F – TEAM TWO LOGS

06-28-2012

daily

Working through understanding the Enigma encryption process over the past week.

5hrs

##############################################################################################

06-28-2012

daily

Joint Application Development session with team. Worked through several examples of Enigma encoding/decoding. Designed front end. Designed class structure for enigma machine.

3.5hrs

##############################################################################################

06-28-2012

daily

Participated in classroom status reports.

1hrs

##############################################################################################

06-29-2012

daily

Prepare pseudocode for the encryption process from the flow identified in last night's JAD session

2hrs

##############################################################################################

07-03-2012

daily

Pair programming with Richard. Setup Rotor class including the 5 rotors and their alphabetic substitutions and two reflectors and their substitutions. Began working on Machine class. Very slow wi-fi issues at ECPI caused the process to work a lot slower than it would normally.

2hrs

##############################################################################################

07-03-2012

daily

Worked on finishing the rotor classes. Began developing the encryption steps. Good through the reflector and into the 3rd rotor coming back.

4hrs

##############################################################################################

07-08-2012

daily

Review code for bugs. Identified and documented several bugs which will be incorporated into code at next pair coding session.

2hrs

##############################################################################################

07-09-2012

daily

Fixed bugs in encryption process. Added classes and code to read in JSON from user and encrypt message. Tested encryption with an online enigma simulator with success.

2hrs

##############################################################################################

07-10-2012

daily

Researching C# solution for dynamic casting of objects. Look at code to debug when rotors are identified by the JSON input.

1hrs

##############################################################################################

07-10-2012

daily

Researched object casting and the dynamic type in C# for the defining of rotors based on json input.

Finished coding plugboard.

Finished coding JSON output.

Deployed web service application to www.koinoniasgr.com/enigma/Encrypt.

Tested rotation, double rotation, different starting points and different rotors in varying slots. Results all match with 3rd party simulator running at enigmaco.de.

5.5hrs

##############################################################################################

07-12-2012

daily

Worked with Oscar on Wed to format JSON for call to webservice.

0.5hrs

##############################################################################################

07-12-2012

daily

Setup database to store and retrieve messages.

0.5hrs

##############################################################################################

07-13-2012

daily

Late entry for Thursday, 7/12. Worked with Richard to fine tune logic for accepting JSON object in web service. Also added functionality to connect to database...not yet complete. Began working with team on Xcode functionality to create JSON object to send to webservice. This took most of the time as this is fairly complex. Was able to generate a JSON object which was later used with Fiddler to access the web service.

5hrs

##############################################################################################

07-13-2012

daily

Debugged and wrote xcode snippets to communicate with enigma machine web service for Oscar and Melony to include in the iPhone app.

1.5hrs

##############################################################################################

07-16-2012

daily

Late Entry for Saturday, 7/14. Created snippets to demonstrate how to read JSON code coming back from Web Service for integration into the XCode applicaiton.

1hrs

##############################################################################################

07-16-2012

daily

Late Entry for Sunday, 7/15. Debugged database error coming back from WinHost account. Had to add a <remove> tag in web.config xml before declaring the database configuration information due to conflict with machine.config.

0.5hrs

##############################################################################################

07-16-2012

daily

Testing and debugging. Added Get method and class for retrieval of messages. Added error checking to ensure that json being received has all the required keys. Completed database writing.

2hrs

##############################################################################################

07-18-2012

daily

Late Entry for Tuesday night. Fixed a bug in the web service caused by the previous evenings error catching code changes. Worked with Oscar to add JSON request and response code to the XCode project. Created a codebook for Melony to begin working with to finalize documentation.

5.5hrs

##############################################################################################

07-18-2012

daily

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

06-28-2012

daily

Research on the function of the enigma machine.

6hrs

##############################################################################################

06-28-2012

daily

Research on the function of the enigma machine.

6hrs

##############################################################################################

07-03-2012

daily

pair programming for 2 hours

research 2 hours

4hrs

##############################################################################################

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

06-26-2012

daily

Out for sick leave. Unable to code in a pair with Jim

5hrs

##############################################################################################

06-26-2012

daily

Reading emails documentation and other team communication.

2.5hrs

##############################################################################################

06-28-2012

daily

Jim gave the summary for our group and he confirmed that a null result is still an result. Other groups gave their summaries as well.

2hrs

##############################################################################################

06-28-2012

daily

Upload all attached email documents to the repository.

1hrs

##############################################################################################

07-02-2012

daily

Created rotor class and began Machine class in C#.

2hrs

##############################################################################################

07-03-2012

daily

Created rotor classes moved items to github added new classes to c# project. Stepped thru classes to confirm output.

4hrs

##############################################################################################

07-09-2012

daily

created rotor class refined how json is entered and what comes out. Also started working on adding multile rotors ( beyond three) with a reflector in any order. Was able to decrypt message using http://enigmaco.de/enigma/enigma.html as a test to encode the message using our app to decode and compare messages.

2hrs

##############################################################################################

07-10-2012

daily

Finished rotors, plugboard, json, testing, deployed to web server.

4hrs

##############################################################################################

07-12-2012

daily

added db connection code improvements worked in xcode to create the json string to send to the web service using a dictionary in a dictionary method to construct the string

3.5hrs

##############################################################################################

07-16-2012

daily

added database functionality, error checking , and GET command to engine.

1.5hrs

##############################################################################################

07-17-2012

daily

finished IOS app, added json send, get and displayed message ppt added to repository iOS app added to repository

5hrs

##############################################################################################

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

06-21-2012

daily

Items covered 6/21/2012:

-set up github, ichat, group log;

-prioritize specifications for Virtual Enigma Simulator;

-Figured out detailed Enigma Rotor/Reflectors inner workings;

-tasked assigned..graphics, diagrams, story, tasks, & test cards;

-more detailed research Enigma logic & design (to include rotors, plug board, and reflectors);

-research how and when the rotors turn and inner workings;

-start power point for presentation

1.5hrs

##############################################################################################

06-21-2012

daily

-update story card;

-develop initial test cards;

-design draft for project power point presentation;

-received the initial diagrams/flow chart for Enigma;

-placed updates to repository

2hrs

##############################################################################################

06-23-2012

daily

-update story cards;

-develop specific test cards;

-set-up blank test cards for projected testing;

-review diagrams (flow chart/logic design) for Enigma;

-develop draft PowerPoint presentation for German Enigma;

-update repository folder with changes/updates;

2.5hrs

##############################################################################################

06-27-2012

daily

-worked on test cards

-worked on status documentation

-worked on manual Enigma encryption

4.5hrs

##############################################################################################

06-28-2012

daily

JAD session

-pairing working on story boards for mobile application

-understand/diagram Enigma encryption path

-pairing hard-copy tested Enigma encryption path

-pairing hard-copy tested Enigma decryption

path

-mapping classes/arrays(rotors, reflectors, plugboard)

-understand/diagram double rotation

-work with iOs diagram

-setup TeamViewer

-get added specifications from client

3.5hrs

##############################################################################################

07-02-2012

daily

-pairing with Oscar to research login/authentication/registration code and alternatives

-pairing with Oscar researching Facebook, Google App Engine, Django/Python, HTTP/HTML/JSON/PHP,

2.5hrs

##############################################################################################

07-11-2012

daily

L/E (late entry)

-pairing on TeamViewer program to code from out of area

-pairing via TeamViewer to code views and interface

-milestone: connectivity issues caused early termination of pairing to code

1.5hrs

##############################################################################################

07-11-2012

daily

L/E (late entry 07-10-2012)

- pairing with Oscar Bernal to work on the following items:

o storyboards and views for Enigma

o set-up interactive keys

o set-up labels (actions and outlets)

o set-up text buttons (actions and outlets)

- Rotors:

o coding keys (26 total keys)

o coding clear button

o coding rotor/reflector to save settings

o coding clear load views

- Json

o discuss json coding strategy for Enigma application

4.5hrs

##############################################################################################

07-11-2012

daily

- pairing with Oscar Bernal to work on the following items:

o coding plugs

o coding json

o coding scheme/look

2.5hrs

##############################################################################################

07-12-2012

daily

- pairing with Oscar Bernal coding for json

o milestone:

• problem with getting return view

o consulting with Dr. Patrick

• communicate with server

• set-up the get json (able to retrieve available messages)

o working on getting 2 arrays with one string in a dictionary

- upload files to GitHub

o milestone:

• lost every piece of documentation except original templates

• schedule tomorrow to re-create documentation

-

3.5hrs

##############################################################################################

07-14-2012

daily

-worked on recreating test cards/story boards

5hrs

##############################################################################################

07-15-2012

daily

-Worked on Team Two paper--APA format

3hrs

##############################################################################################

07-17-2012

daily

o Pairing with Oscar Bernal to code in Enigma

o Worked on login/authentication via Facebook

• Tested login/authentication via Facebook

• Tested accessing application after login to Facebook

4.5hrs

##############################################################################################

Fixed a bug in the XCode project which was preventing text from being parsed if no encryption was done. Problem introduced in the web service when error catching changes were made on Monday evening.

1hrs

##############################################################################################

APPENDIX G – CODE BOOK

qwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmrtyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnmqwertyuiopasdfghjklzxcvbnm

|  |
| --- |
| Enigma Machine Application  Team Two: Code Book  7/18/2012  MHenderson |

Enigma Machine Application Code Book

“A Cloud-Based Working Simulation for Modern Cryptography”

Oscar Bernal, Jim Esposito, Melony Henderson, Richard Robinson

Computer Information Systems Graduate

Program Night Students

ECPI University

|  |
| --- |
| Code Book Legend (Codebuch Legende): |
|  |
| Schlüssel key |
| Walzenlage rolling position |
| Ringstellung ring position |
| L left rotor |
| C center rotor |
| R right rotor |
| Steckerverbindungen plug connections |
| Rückstrahler reflector |
|  |

***Ringstellung***

**01 02 03 04 05 06 07 08 09 10 11 12 13**

**A B C D E F G H I J K L M**

**14 15 16 17 18 19 20 21 22 23 24 25 26**

**N O P Q R S T U V W X Y Z**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Schlüssel | | Walzenlage | | Ringstellung | Steckerverbindungen | Rückstrahler |
|  | L C R | | L C R | |  | Select B or C |
| 31 | V I II | | 24 05 22 | | A:Y,B:N,C:E,D:V,F:H,G:W,K:U,L:Q,M:P,R:T | - - |
| 30 | II V III | | 04 06 19 | | B:R,C:N,D:J,E:X,G:U,H:W,I:L,K:Z,M:T,O:Q | - - |
| 29 | I III II | | 04 01 03 | | A:Z,B:H,C:O,D:J,E:R,G:S,K:X,M:P,Q:T,W:Y | - - |
| 28 | III V IV | | 13 21 02 | | A:F,B:J,D:Q,E:I,G:O,K:L,M:S,N:U,P:T,V:X | - - |
| 27 | IV V III | | 07 04 17 | | A:U,D:L,E:I,F:T,G:Y,H:J,K:W,M:O,P:R,V:Z | - - |
| 26 | I IV V | | 19 19 25 | | B:O,C:U,D:E,F:V,G:X,I:Z,J:Q,K:L,M:T,R:Y | - - |
| 25 | I IV II | | 02 14 05 | | A:X,B:M,C:Z,D:S,E:U,F:Q,H:P,L:V,N:Y,R:T | - - |
| 24 | IV I V | | 21 04 14 | | B:G,C:L,E:F,H:K,J:O,M:P,N:T,R:Y,S:W,U:X | - - |
| 23 | I V II | | 17 08 04 | | B:M,C:O,D:P,E:G,F:U,J:Y,L:N,Q:R,T:Z,V:X | - - |
| 22 | I IV III | | 18 08 18 | | B:S,C:X,D:I,E:V,F:W,H:Q,J:L,K:Z,P:T,R:U | - - |
| 21 | I V IV | | 01 08 24 | | A:U,C:V,D:M,E:I,F:K,G:Y,H:L,J:S,O:X,P:Q | - - |
| 20 | V I IV | | 01 17 09 | | A:K,B:O,C:S,D:T,E:Z,G:X,H:I,J:V,L:U,N:Y | - - |
| 19 | II III IV | | 21 04 14 | | A:Z,B:J,C:X,D:Y,E:F,I:M,K:O,L:T,P:U,R:V | - - |
| 18 | IV V I | | 04 16 11 | | A:M,C:S,D:L,E:K,F:P,G:J,H:U,I:Z,Q:T,R:Y | - - |
| 17 | V III II | | 23 11 05 | | A:C,B:D,F:N,G:W,H:U,I:X,J:K,L:M,P:V,R:S | - - |
| 16 | IV I III | | 07 15 08 | | A:H,B:G,C:J,E:S,K:Z,M:O,N:V,P:W,Q:R,U:X | - - |
| 15 | III I II | | 07 07 23 | | B:N,C:H,D:Z,E:F,G:R,K:P,L:X,M:T,Q:U,V:W | - - |
| 14 | II I V | | 11 25 16 | | A:O,D:V,E:N,F:I,G:R,H:P,K:T,L:S,M:W,Q:U | - - |
| 13 | IV V I | | 23 23 02 | | B:F,D:T,E:Q,G:J,I:V,K:P,L:X,M:U,N:Z,R:Y | - - |
| 12 | IV I II | | 10 16 05 | | A:X,B:C,D:Y,E:R,G:S,I:P,L:W,M:Z,N:O,Q:V | - - |
| 11 | I IV II | | 01 01 25 | | A:T,B:J,C:N,E:P,H:Z,K:V,L:O,M:S,Q:U,X:Y | - - |
| 10 | I V IV | | 07 20 20 | | A:Y,B:K,C:S,D:U,E:R,F:G,I:L,J:Q,M:Z,O:W | - - |
| 9 | III I II | | 05 10 23 | | B:D,C:J,E:Z,F:O,G:T,I:U,L:N,Q:S,V:Y,W:X | - - |
| 8 | I IV II | | 04 22 13 | | B:Q,D:S,E:Y,F:M,G:Z,H:J,I:V,K:L,O:R,U:X | - - |
| 7 | III V I | | 10 23 21 | | A:V,B:Y,D:U,F:K,G:N,J:T,L:R,M:W,O:X,S:Z | - - |
| 6 | II III V | | 16 13 04 | | A:V,B:P,C:H,F:J,I:U,K:Z,L:O,N:S,Q:W,T:Y | - - |
| 5 | I III IV | | 14 26 24 | | A:Q,B:D,C:L,E:M,F:J,H:Z,K:Y,N:U,O:R,S:V | - - |
| 4 | IV III I | | 25 22 16 | | A:U,C:O,E:K,F:H,I:Q,J:V,L:R,M:N,S:Z,T:W | - - |
| 3 | II I III | | 17 15 16 | | A:K,B:J,C:V,D:L,E:R,G:Q,H:M,I:O,T:Y,X:Z | - - |
| 2 | IV V III | | 24 03 05 | | A:F,B:Z,D:R,E:N,G:P,H:V,I:X,J:Y,K:Q,O:T | - - |
| 1 | I IV V | | 01 15 19 | | A:J,D:M,F:P,G:Z,I:T,K:U,L:R,N:X,O:Q,V:W | - - |
| 0 | V III IV | | 21 23 15 | | <enter up to 10 plugs (Steckerverbindungen)> | - - |
| -1 | III II I | | 25 10 08 | | <enter up to 10 plugs (Steckerverbindungen)> | - - |
|  |  | |  | |  |  |

(Rijmenants, 2004-2012)

Definitions

**Cipher:** a method of secret writing using substitution or transposition of letters according to a key.

**Cryptography:** the science or study of secret writing, especially code and cipher systems.

**Decoding:** converting from code into plain text.

**Decryption:** to decipher; to decode.

**Encoding:** to put (a message, for example) into code.

**Encryption:** to put into code or cipher.

**Enigma Machine:** an electro-mechanical rotor machine; cipher machine used to encrypt and decrypt secret messages.

**Extreme agile software development:** a group of software development methodologies based on iterative and incremental development, where requirements and solutions evolve through collaboration between self-organizing, cross-functional teams.

**Pair programming:** an agile software development technique in which two programmers work together at one work station. One types in code while the other reviews each line of code as it is typed in. The person typing is called the driver. The person reviewing the code is called the observer (or navigator). The two programmers switch roles frequently.

**Software engineering:** the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches

# References

Carlson, A. (1997-2004). *Enigma Simlulator*. Retrieved JUN 28, 2012, from BLETCHLEY PARK National Code Centre: http://www.bletchleypark.org.uk/content/enigmasim.rhtm#

Farlex, I. (2012). *The Free Dictionary by FARLEX*. Retrieved JUL 18, 2012, from www.the freedictonary.com: http://encyclopedia.thefreedictionary.com/

Google. (Not Available). *Google Translate*. Retrieved JUL 18, 2012, from translate.google.com: http://translate.google.com/#en|de|code%20book%20legend

Long, T. (2009, AUG 31). *Enigma Simulator v.1.2.4*. Retrieved JUN 28, 2012, from Enigma Simulator by Terry Long: http://www.terrylong.org/

Miller, D. A. (1996-2001). *The Cryptographic Mathematics of ENIGMA.* Retrieved JUL 19, 2012, from Center for Crypologic History, National Security Agency : http://ed- thelen.org/comp-hist/NSA-Comb.html

Psaul Reuvers & Marc Simons. (2009-2012, JUN 16). *Enigma Cipher Machine*. Retrieved JUN 26, 2012, from Crypto Museum [cryptomuseum.com]: http://www.cryptomuseum.com/crypto/enigma/

Rijmenants, D. (2004-2012). *Enigma Details*. Retrieved JUN 27, 2012, from Technical Details of the Enigma Machine: http://users.telenet.be/d.rijmenants/en/enigmatech.htm

Schwager, R. (1998-2005). *The Enigma Machine*. Retrieved JUN 28, 2012, from Enigma Machine Applet: http://russells.freeshell.org/enigma/

Sommerville, I. (2001). *Software Engineering, ninth edition.* Boston: Addison-Wesley.

Spiess, F. (2009). *EnigmaCo.de (Enigma Simulation)*. Retrieved JUN 28, 2012, from Three Rotor Enigma Simulation (Enigma Simulation Test Site): http://enigmaco.de/enigma/enigma.html